SCHEMA III

USERS MANUAL

OMATION

SCHEMA III Version 3.31
SUMMARY OF ENHANCEMENTS

SCHPOST:

The previous versions of the post-processor held all the pages in memory and processed them in the 640K. There was a limit to how many pages could be processed. This new processor compiles each top level schematic page into its own pinlist object file with an 'O' extension. These files are merged and sorted to create a parts file .PST and a nets file .NST. If one of the pages of your schematic is changed, then SCHPOST will do an Incremental Compile. Where only that one page will be reprocessed and then the pinlists are merged and sorted again. If you have not changed the schematic at all, then no processing will be done and the .PST and .NST files will be used again. This new approach will yield no memory limits across pages since the sorting is being done on disk. If you have the disk space you can create the netlist. The only limit you can reach is if one particular top level page contains too deep a hierarchy.

LABEL SWAPPING:

Enclosed with the SCHEMA is a library part called, **LBLSWAP**. This part will allow the user to substitute one net name for another upon processing.

SCHPRINT/SCHPLOT:

Supports the three new title block variables **<FT>**, **<FP>**, and **<FF>**. (see SCHDRAW)

SCHDRAW:

Auto-Backup

A backup of the current page you are working on is sent to a file, **BACKUP.P01**. It is output to the directory path assigned by the **OMATION** environment variable or the current directory. Default is five minutes.

Title Block variables

Three automatic variables are available to be placed in your title block.

Use **NOTES-TEXT** for placement (see your manual)

<FT> - Time and Date Stamp
<FP> - Shows the current page number ("1" of 6)
<FF> - Path and filename

Vga Switch

If using a high resolution graphics adapter and you want to see more pixel definition then you can select, **VGA** and the graphics will be switched.

Graphics Adapters

The Trident Impact III series is now supported. The following are the key letters to be placed in the **GRAPH=** variable of the SCHEMA.CFG or use the new SCHCFG.EXE:

I1 - 800 x 600
I2 - 1024 x 768

Updating Config within SCHDRAW

All the parameters can be changed from within SCHDRAW and saved to a config file. If you select **CONFIG-LIBR 1,** type in a new library , and press **<Enter>,** the library can be used. Library browse can be invoked by typing : ***.LBR <Enter>** Select the library of choice with the mouse or cursor keys. **COLORS** and **PAN** are saved to the **SCHEMA.CFG** all others are saved to whatever **DRAWING.CFG** is loaded.

Auto Pin Creation

The **BODY-PIN** command in the Object Editor will let the user create normalor bubbled; horizontal or vertical pins with definitions. The header will have the definition attached. The Body text is placed to the right of the header. The pin definition is grouped and ready to place. The hidden pins are assigned by using the '@' icon. Note when you place the pin that the pieces are now separate. If you use **EDIT-MOVE,** each individual piece will move. If you select **EDIT-REGION,** however, a small region encloses the definiton.

Regions

Regions now dynamically rubberband the wires. Also, once a region is defined it does not pickup extra items that may be found in it after a **MOVE.** If the commands **MOVE** or **COPY** are selected again the contents of the region are re-defined. Multiple **REGION-COPYs** when **UNDO** is selected undoes only the last copy.

Undo

Undo toggles between the last two commands. If you Undo an **EDIT-MOVE** accidentally select **UNDO** again and Undo undoes itself.

EDIT-WIRE

This command will allow the user to adjust the wires of an existing part or one that was rubberbanded. Both endpoints of a wire remain the

same. However, if a wire is selected and move a new wire is created. Thus by working from both ends of a wire, a new wire path can be created.

FILE-PRINT

Most users print in foreground, thus, when this command is selected the default output is **PRN**. If **PRINT** is selected again then the file with the .Txx extension is displayed for background printing.

MAXIMUM OBJECT SIZE

Users can create an object limited only by the Object Editor window boundaries. However, if the object is too large, then you may have to place the part in the schematic using the 2:1 zoom mode.

MISCELLANEOUS

- Improved memory management has freed 4.7 K for the page buffer

- The "Object Not Found" message is generated if:
    1. drawing being loaded
    2. libraries modified using **CONFIG-LIBR**
    3. Leave Object Editor to Schematic Editor
    4. Object is entered using **DRAW-OBJECT, EDIT-IMAGE,** or **LIBR-LOAD**

This feature allows page items to be copied or moved without generating the same "Object Not Found" error messages.

-New common commands **DRAW-ERASE, NOTES-MOVE, NOTES-COPY,** and **NOTES-ERASE** reduce the need to select new primaries for basic operations.

SCHEMA III PACKAGE CONTENTS

SCHEMA III consists of the following items:

1 Omation License Agreement
1 SCHEMA III User's Manual
2 SCHEMA III PROGRAM Disks (1.2 Meg)

Ensure that you have received these items. If something is missing, please contact us.

Please complete and return the Omation Registration Card enclosed with your package to facilitate Customer Support.

SCHEMA III

USERS MANUAL

OMATION

THIS AGREEMENT CONSTITUTES THE COMPLETE AGREEMENT BETWEEN YOU AND OMATION INCORPORATED. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT OPEN THE DISK PACKAGE. PROMPTLY CALL OMATION INCORPORATED AT (800) 553-9119 FOR A RETURN AUTHORIZATION NUMBER (RMA). UPON RECEIPT OF THIS AUTHORIZATION, RETURN THE FULL PACKAGE (INCLUDING MANUALS AND BINDERS) TO OMATION WITHIN THREE (3) DAYS POSTPAID FOR DELIVERY WITHIN TEN (10) DAYS FOR A REFUND.

OMATION LICENSE AGREEMENT

1. DEFINITIONS:

A. "Omation" shall mean Omation Incorporated, 801 Presidential Drive, Richardson, TX 75081 which is the owner of this program.

B. "Licensee" shall mean the end user of this product, either individual or company as it is shown on the books of Omation.

C. "SCHEMA III" shall mean the software and related manuals and materials which are part of this package.

D. "Computer" shall mean the computer on which this product is used.

E. "License" shall mean this AGREEMENT and the rights and obligations created hereunder by the United States Copyright Law and Texas law.

2. LICENSE:

Omation grants to Licensee a non transferrable, non exclusive license to use this copy of SCHEMA III on any single computer at a single location as long as Licensee complies with the terms of this LICENSE. If Licensee violates the terms of this License, Omation may, at its option, terminate this LICENSE. LICENSEE agrees that LICENSOR is and shall remain sole owner of the software. In the event of such termination, LICENSEE agrees to return SCHEMA III to Omation. SCHEMA III is the sole and exclusive property of Omation. LICENSEE ownership is limited to the diskette(s) purchased.

3. COPIES

LICENSEE agrees to make no more than two (2) copies of SCHEMA III for backup and archival purposes. LICENSEE agrees to include the copyright notice on each copy and to label copies with all information included on the original diskette label. In the event that LICENSEE uses the original diskette and/or any copy or copies simultaneously on different computers, LICENSEE agrees to request from Omation and pay for licenses for the appropriate number of additional user copies. By the act of opening the packs in which the diskettes are sealed, LICENSEE subscribes to and agrees to the terms of this License.

4. TERM, TRANSFER AND RESTRICTIONS

THIS SOFTWARE PROGRAM OR ANY COPY THEREOF MAY NOT BE MODIFIED, TRANSFERRED, USED, COPIED, NOR MERGED IN WHOLE OR IN PART EXCEPT AS EXPRESSLY PROVIDED FOR IN THIS LICENSE.

The SCHEMA III software may be used only on one computer at time. The program may be transferred to another computer provided that SCHEMA III is used in connection only with one computer at a time. SCHEMA III may not be transferred electronically from one computer or file system to another computer over a network. This system and its documentation may not be modified without express written permission of Omation.

Licensee agrees to take reasonable caution and exercise due diligence in protecting SCHEMA III from unauthorized distribution, disclosure, reproduction or publication. Unauthorized distribution and/or reproduction of these materials may be a crime thus subjecting Licensee to civil and criminal prosecution. Licensee may not transfer any copy of SCHEMA III to any other person or company without the express prior written permission of Omation. Omation reserves the right to revoke this License and/or to seek other legal remedies to which it might be entitled should these conditions be violated.

5. LIMITED WARRANTY

SCHEMA III is provided "as is" without warranty as to performance, merchantability, or fitness for any particular purpose. The entire risk as to suitability, results, and performance rests with Licensee.

However, to Licensee only and provided Licensee has returned to Omation the signed registration and warranty card, Omation warrants the magnetic diskette(s) on which the program is provided to be free from defects in materials and faulty workmanship under normal use for a period of ninety (90) days from the date of purchase . If the diskettes fail due to neglect, accident, or abuse, Omation shall not be liable to replace the diskette(s) under this Limited Warranty.

Licensee's sole and exclusive remedy in the event of a defect is expressly limited to the replacement of the diskette(s) as provided above.

Any implied warranties relating to the diskette(s) including any implied warranties of merchantability and fitness for a particular purpose are limited to a period of ninety (90) days from the date of purchase. Omation in no event shall be liable for incidental, indirect, special or consequential damages resulting from the use of this product, including lost revenues and profits or loss of use or other reasonable loss, whether the cause of such damages was known to Omation or not and in no event shall Omation's liability arising in connection with SCHEMA III exceed an amount equal to the purchase price of SCHEMA III. Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitations might not apply to Licensee. This warranty gives licensee specific legal rights and licensee may also have other rights which vary from state to state. Licensee and Omation agree that SCHEMA III is not intended as "Consumer Goods" under state or federal warranty laws.

6. PROGRAM UPDATES

Program updates are available to registered users only. If the Omation registration card has not been received by Omation, program updates need not be made available. Omation continues a program of product improvement. With evidence of purchase date, Omation will provide Licensee with free upgrades and improvements for ninety (90) days following the date of purchase.

7. MISCELLANEOUS

Omation maintains a policy of a full thirty day money back guarantee of SCHEMA III. Provided Licensee has called Omation and secured a Return Materials Authorization (RMA) within thirty days of purchase and, upon receipt of RMA, ships all materials including diskettes, program manuals, binders, and other components comprising SCHEMA III insured and postpaid to Omation within three (3) days of receipt of RMA, Licensee shall be entitled to a refund of purchase price less initial shipping and handling charges.

In the event of legal action brought by either Omation or Licensee to enforce the terms of this licensing agreement, the prevailing party shall be entitled to recover reasonable attorney's fees and expenses in addition to other relief deemed appropriate by the trial court or any appellate court. Jurisdiction for all disputes shall be exclusively in Texas which jurisdiction Licensee hereby submits to. This AGREEMENT shall be governed by the laws of the State of Texas.

This agreement shall represent the only agreement between Licensee and Omation. It may not be modified by any representation, written or verbal, without the express written authorization and consent of a corporate officer of Omation.

SCHEMA III

REFERENCE MANUAL

Copyright, OMATION, Inc. 1991

OMATION, Inc.
801 Presidential
Richardson, TX 75081

SUPPORT: 1-800-562-9085
SALES: 1-800-553-9119
FAX: 1-214-783-9072
BBS: 1-214-231-1350

Revision: 33-A

Information contained herein is subject to change without prior notification. The software described in this user's manual has been supplied under license or non-disclosure aggreement and may only be used in accordance with the terms of the agreement. Omation, Inc. provides this manual without warranty of any kind, expressed or implied. This user's manual may contain technical and/or typographical errors.

Copyright Omation, Incorporated, 1991

Omation , the Omation logo, SCHEMA III and SCHEMA III logo are trademarks of Omation, Incorporated.

Apple and Apple Imagewriter are trademarks of Apple Computer, Incorporated / AT&T and AT&T 6300 are trademarks of American Telephone and Telegraph / AutoCAD and Drawing Exchange Format are trademarks of Autodesk, Inc. / Cadnetix is a trademark of Cadnetix Corp. / Calay is a trademark of Calay Systems, Inc. / Canon and Canon LBP-8 A2 are trademarks of Canon USA, Inc. / Citizen, Citizen MSP10 and MSP15 are trademarks of Citizen America Corp. / COMPAQ, COMPAQ PORTABLE, COMPAQ DESKPRO, and COMPAQ 386 are trademarks of COMPAQ COmputer Corp. / Computervision is a trademark of Computervision Inc. / dataCon is a trademark of dataCon, Inc. / dBase III is a trademark of Ashton-Tate Corp. / EE Designer and Visionics are trademarks of Visionics Corp. / Epson is a trademark of Epson Corp. / MX100, RX100, FX100, FX100+, FX185, GRAFTRAXPLUS, MX80, RX80, RX100, FX100, FX100+, FX185 are trademarks of Epson America Inc. / FutureNet is a trademark of Data I/O Corp. / Genicom 3000 is a trademark of Genicom Corp. / Genoa Super EGA, and GENOA Super EGA Hi Res are trademarks of Genoa Corp. / Hercules is a trademark of Hercules Computer Technology / Hewlett-Packard, HP, HP-GL, HP7475, HP7470, HP Laserjet+ and HP

Vectra are trademarks of Hewlett-Packard Co. Houston Instruments PC695/595, DMP40/41/50/51 are trademarks of Houston Insruments . IBM, IBM PC/XT/AT/Portable PC, IBM Color Graphics Adapter, IBM Multi-Color Graphics Array, IBM Enhanced Graphics Adapter, IBM Video Graphics Array, Standard Color Graphics Adapter, Color Printer and ProPrinter are trademarks of International Business Machines Corp. / Intergraph is a trademark of Integraph / Logitech is a trademark of Logitech Inc. / LOTUS and Lotus 1-2-3 are trademarks of Lotus Development Corp. / Microsoft and MS-DOS are trademarks of Microsoft Corp. / Mouse Systems and PC Mouse are trademarks of Mouse Systems Corp. / NEC and NEC MutiSync are trademarks of NEC Home Electronics (USA) Inc. / Okidata and Okidata 82A/83A/ML92/ML93/192/193/2410 are trademarks of Okidata Corp. / Paradise AutoSwitch EGA 480 and VGA Plus are trademarks of Paradise Corporation / P-CAD is a trademark of Personal CAD systems, Inc. / PostScript is a trademark of Adobe Systems Inc. / Printroni 300/600 is a trademark of Printronix / Racal Redac is a trademark c Racal Redac, Inc. / SCI-CARDS is a trademark of Scientific Calculations/ Sigma Designs and Sigma Color 400 are trademarks of Sigma Designs, Inc. / Star is a trademark of Star Micronics / Summagraphics is a trademark of Summagraphics Corp. / Taxan and Taxan 710 are trademarks of Taxan Corp. / Tecmar and Tecmar Graphics Master are trademarks of Tecmar / Telesis is a trademark of Telesis Systems Corp. / Texas Instruments 850 is a trademark of Texas Instruments / Torrington and Manager Mouse are trademarks of Torringtom / Toshiba and Toshiba P340/350/1340/1351/T3100 are trademarks of Toshiba America, Inc. / Video - 7 is a trademark of Video Seven Inc. / Wyse and Wyse WY-700 are trademarks of Wyse Technology / Zenith is a trademark of Zenith Electronics Corp.

Table of Contents

Chapter 1. Getting Started 1
1.1 INTRODUCTION 1
1.2 SYSTEM REQUIREMENTS 3
1.3 DISK BACKUPS 5
1.4 INSTALLATION 6
1.5 TUTORIAL - SCHEMA III MADE SIMPLE 6
Using a Mouse with SCHEMA III 9
Using the Keyboard with SCHEMA III 10
Auto Backup 10
BEGINNING THE PROGRAM 11
LESSON 1 - AN OVERVIEW OF SCHDRAW 12
SCHDRAW HELP 12
SCHDRAW ZOOM 13
SCHDRAW - DRAW 15
SCHDRAW EDIT 21
SCHDRAW NOTES 23
SCHDRAW FILE 27
SCHDRAW CONFIG 29
SCHDRAW SEARCH 30
CONCLUSION 31
LESSON 2 - USING THE MANAGER AND SCHCFG 33
1.5.2 The SCHEMA Program Manager. 33
1.5.3 SCHCFG 37
LESSON 3- USING SCHDRAW TO CREATE A SCHEMATIC 43
1.5.4 SCHDRAW 43
1.5.4.3 The Schematic Editor 46
1.5.4.3.1 BROWSING the Libraries 52
LESSON 4- CREATE A PART USING THE OBJECT EDITOR 61

1.5.4.4 The Object Editor 61
1.5.4.4.1 What is a Name Object? 62
1.5.4.4.2 Body Objects 67
1.5.4.4.3 Discrete Name and Body Objects 74
1.5.4.4.4 Nested Objects 81
1.5.4.4.5 The BODY-BIT Command 82
1.5.4.4.6 The EDIT-ID Command 84
1.5.4.4.7 Remaining Object Editor Commands 85
1.5.4.4.8 Creating the Tutorial Component 85
1.5.4.4.8.1 Tutorial Body Object 87
Tutorial Name Object 93
LESSON 5- FINISH THE LOGIC SCHEMATIC 97
1.5.4.5 Completing the LOGIC Schematic 97
1.5.4.6.1 Advantages of Hierarchically Structured Drawings 115
1.5.4.6.2 Creating the Tutorial Hierarchical Drawing 115
1.5.4.6.2.1 The Tutorial Hierarchical Body Object 115
1.5.4.6.2.2 The Tutorial Hierarchical Name Object 118
1.5.4.6.2.3 The Tutorial Hierarchical Drawing 123
1.5.4.6.3 Using the PUSH\POP Feature 130
LESSON 7- Using SCHPOST 133
1.5.4.7 SCHPOST 133
LESSON 8- Using SCHPLOT & SCHPRINT 145
1.5.4.8 SCHPLOT/SCHPRINT 145
1.5.4.8.1 Invoking the SCHPLOT Tutorial 145
1.5.4.8.2 Invoking the SCHPRINT Tutorial 145
LESSON 9- Using SCHLIB and Library Management 148
1.5.4.9 SCHLIB 148
1.5.4.9.1 Invoking the SCHLIB Tutorial 148
1.5.4.9.2 Library Philosophy 155
1.5.4.9.3 Assign a User Library 155
1.5.4.9.4 Archive a Project 156

1.6 THE SCHEMA-PCB TUTORIAL 157
1.7 WHAT NEXT? 157

Chapter 2. SCHEMA Program Manager 1
2.1 INTRODUCTION 1
2.2 INVOKING THE MANAGER 1
2.3 RUNNING THE MANAGER 2
2.4 COMMAND DESCRIPTIONS 6
2.4.1 Manager Control Commands 6
2.4.2 SCHEMA III Commands 12
2.4.3 SCHEMA-PCB Commands 23
2.5 SCHEMA Manager Configuration 27
2.5 SCREEN DESCRIPTION 28
2.6 CUSTOMIZING THE MANAGER 28
2.7 CONFIGURATION FILE FORMAT 32
2.7.1 COLORS 37
2.7.2 COMMAND LINE VARIABLES 38
2.7.3 EXECUTION PARAMETERS 40

Chapter 3. SCHDRAW 1
3.1 INTRODUCTION 1
3.2 INVOKING SCHDRAW 2
3.2.1 SCHDRAW Screen Format 3
3.2.2 Moving the Cursor 5
3.2.3 Selecting Commands 6
3.2.3.1 Using the Cursor to Select Commands 6
3.2.3.2 The Middle Mouse Button 8
3.2.3.3 Using the Keyboard to Select Commands 8
3.2.3.4 Auto Backup 9
3.2.4 Editing With a Mouse/Keyboard 9
3.2.5 UNDO 9
3.3 USING THE SCHDRAW SCHEMATIC EDITOR 10
3.3.1 Blank Pages 10

3.3.2 The Drawing Title Block 12
3.3.3 Object Libraries 14
3.3.4 Basic Schematic Drawing Techniques 16
3.3.4.1 Drawing Aids 16
3.3.4.2 Using Components 20
BROWSE 20
3.3.4.2.1 How to Select Different Component Gate Sections, Rotated Orientations, and DeMorgan Equivalents 21
3.3.4.2.2 Reference Designators 23
3.3.4.2.3 Assigning Other Parameters to Components 24
3.3.4.3 Making Connections for SCHPOST 26
3.3.4.3.1 Signal Connections 26
3.3.4.3.2 Label Connections 27
3.3.4.3.3 Bus Connections 34
3.3.5 Printing/Plotting SCHDRAW Drawings 35
3.4 USING THE SCHDRAW OBJECT EDITOR 39
3.4.1 Invoking the Object Editor 39
3.4.2 Types of Objects 39
3.4.3 Name Objects 43
3.4.4 Body/Nested Objects 47
3.4.4.1 Assigning Power and Ground 49
3.4.4.2 Referencing Parameters Within BODY-PIN Entries 51
3.4.4.3 Making Connectors 51
3.4.5 Hierarchical Objects 53
3.4.6 Maximum Body Object Size 54
3.4.7 Object Naming Conventions 54
3.5 COMMAND DESCRIPTIONS 55

Chapter 4. SCHPLOT/SCHPRINT 1
4.1 SCHPLOT 1
4.1.1 Introduction 1
4.1.2 Plotter Communications 1

4.1.3 Invoking SCHPLOT 3
4.1.4 Running SCHPLOT 11
4.1.5 PLOTTER PROBLEM CHECKLIST 12
4.1.6 HEWLETT-PACKARD (HP) 14
4.1.7 HOUSTON INSTRUMENT (HI) 14
4.1.8 CALCOMP (CP) 15
4.1.9 POSTSCRIPT (PS) 16
4.1.10 DRAWING EXCHANGE FORMAT (DXF) 17
4.1.11 Customizing Pages For A Particular Plotter 18
4.2 SCHPRINT 20
4.2.1 Introduction 20
4.2.2 Invoking SCHPRINT 21
4.2.3 Running SCHPRINT 25
4.2.4 Customizing Pages For A Particular Printer 25
4.2.5 Interfacing to Word Processors using TIFF 26

Chapter 5. SCHCFG 1
5.1 INTRODUCTION 1
5.2 CONFIGURATION FILE FORMAT 1
5.2.1 Variable Description 2
5.3 INVOKING SCHCFG 4
5.4 RUNNING SCHCFG 5
5.5 COMMAND DESCRIPTIONS 15

Chapter 6. SCHLIB 1
6.1 INTRODUCTION 1
6.2 OBJECT LIBRARY COMPOSITION 1
6.3 INVOKING SCHLIB 2
6.4 RUNNING SCHLIB 3
6.4.1 Main Menu Library Operations 6
6.4.2 Main Menu General Object Operations 7
6.4.3 Name Object Editor Menu 8

6.4.4 Compressing Libraries and Expanding Directories 8
6.4.5 Library Philosophy 9
6.4.6 Assign a User Library 9
6.4.7 Archive a Project 11
6.5 COMMAND DESCRIPTIONS 11

Chapter 7. SCHPOST 1
7.1 INTRODUCTION 1
7.2 POST-PROCESSORS 2
7.2.1 Routing 2
7.2.2 Bill of Materials 4
7.2.3 Design Rule Check 5
7.2.4 Net List 6
7.2.5 Pin List 6
7.2.6 Wire List 7
7.2.7 Reports 7
7.3 OPERATION 7
7.3.1 Invoking SCHPOST 7
7.3.2 Running SCHPOST 10
7.4 OUTPUT CONFIGURATIONS 13
7.4.1 Routing 13
7.4.2 Bill of Materials 15
7.4.3 Design Rule Check 23
7.4.4 Net and Pin Lists 27
7.4.5 Wire List 27
7.4.6 Reports 30
7.4.7 Label Swapping 33
7.5 HIERARCHY 34
7.5.1 Hierarchical Design Guidelines 39
7.5.2 Global Nets 40
7.5.2.1 Global Net Guidelines 40
7.5.3 Multiple Configuration Files 41
7.6 CONVERSION PROGRAMS 41

7.7 VIRTUAL MEMORY 43

Chapter 8. Utilities 1
8.1 INTRODUCTION 1
8.2 SCHBACK 1
8.2.1 Invoking SCHBACK 4
8.2.2 SCHBACK Tutorial 6
8.2.3 SCHBACK User Messages 6
8.3 SCHECO 7
8.3.1 Invoking SCHECO 7
8.3.2 Running SCHECO 8
8.4 SCHLOCAL 9
8.4.1 Invoking SCHLOCAL 10
8.4.2 Running SCHLOCAL 10
8.5 SCHPAGE 11
8.5.1 Invoking SCHPAGE 12
8.5.2 Running SCHPAGE 13
8.6 NET/PIN LIST CONVERSION PROGRAMS 14
8.6.1 Net List Suggestions 16
8.6.2 Using SCHEMA III With PSpice (Analog) 16
Using PSpice for Digital/Analog Applications 23
8.7 Network Capatibility 27

Appendix A. Installing SCHEMA III 1
A.1 INTRODUCTION 1
A.2 SCHEMA III DISK CONTENTS 1
A.3 MANUALLY INSTALLING SCHEMA III TO ON HARD DISK SYSTEMS 3
A.4 CONFIGURING SCHEMA III TO WORK WITH YOUR HARDWARE 5

ppendix B. Net/Pin List Formats 1
B.1 INTRODUCTION 1

B.2 NET LIST FORMAT 1
B.3 PIN LIST FORMAT 4
B.4 NET/PIN LIST CONVERSION PROGRAMS 8

Appendix C. Keyboard Usage 1
C.1 INTRODUCTION 1
C.2 TEXT EDITING KEYS 1
C.3 SCHDRAW FUNCTION KEYS 2
C.4 PARTS BROWSING FUNCTION KEYS 4
C.5 INTERNATIONAL CHARACTERS 5

Appendix D. Error Messages 1
D.1 INTRODUCTION 1

Appendix E. Hardware Requirements 1
E.1 INTRODUCTION 1
E.2 PC 1
E.3 GRAPHICS ADAPTER 2
E.4 MOUSE 10
E.5 PRINTER 11
E.6 PLOTTER 24

Appendix F. Object Libraries 1
F.1 THE STANDARD COMPONENT LIBRARY 1
F.2 GRAPH.LBR: THE GRAPHICS LIBRARY 6
F.3 IEEE.LBR: THE IEEE/IEC COMPONENT LIBRARY 6
F.4 THE ROTATED / GATE PARTS 7

Chapter 1. Getting Started

1.1 INTRODUCTION

SCHEMA III is a highly effective schematic processing program for electrical engineers, technicians, draftsmen, and other engineering professionals. Use SCHEMA III to quickly generate schematic diagrams, wire lists, block diagrams, timing diagrams, etc.

SCHEMA III is also a Program Manager which can organize all the software stored on your hard disk. By creating a menu-style format, your ability to quickly access different programs (such as a word processor, database, line editor, and of course, the SCHEMA III schematic diagrams) is greatly enhanced.

SCHEMA III is designed to be used with the IBM PC/XT/AT/386 family of personal computers. Paragraph 1.2, *System Requirements*, and Appendix E, *Hardware Requirements*, cover the specific hardware required/supported by SCHEMA III.

This manual explains how to use SCHEMA III and has been written with the assumption that you are familiar with basic DOS functions. If you are not familiar with DOS, please consult your DOS user's manual before using SCHEMA III.

Chapter 1 provides an overview of SCHEMA III and outlines the procedure to install and configure the program. A tutorial will guide you through both the basics of SCHEMA III's schematic functions, and

explain how to customize the Program Manager to fit your needs. This chapter is 'must' reading before trying to use SCHEMA III.

Chapter 2 describes in greater detail the SCHEMA Program Manager. In essence, your entire operating system is converted to an easy-to-understand menu.

Chapter 3 describes SCHEMA III's SCHDRAW program and contains a summary of all the SCHDRAW commands. SCHDRAW is the heart of SCHEMA III's schematic functions, providing the graphics tools to create drawings.

Chapter 4 contains information on printing and plotting your SCHEMA III drawings with the SCHPLOT and SCHPRINT programs.

Chapter 5 covers SCHCFG, SCHEMA III's configuration manager.

Chapter 6 describes SCHLIB, SCHEMA III's object library manager.

Chapter 7 describes SCHEMA III's post-processing program, SCHPOST.

Chapter 8 describes SCHEMA III's utility programs, which includes **SCHBACK, SCHECO, SCHPAGE, CONVERT, PSpice,** and **Networks.**

Appendix A covers the manual installation of SCHEMA III.

Appendix B covers SCHEMA III's net and pin list formats.

Appendix C explains SCHEMA III's text editing keyboard usage and conventions.

Appendix D contains a list of SCHEMA III's error messages.

Appendix E covers SCHEMA III's hardware requirements.

Appendix F contains some descriptions of objects contained in SCHEMA III's object libraries.

1.2 SYSTEM REQUIREMENTS

SCHEMA III requires the following:

IBM PC/XT/AT/386 or compatible PC with 512K RAM minimum (640K recommended)

DOS 3.0 or above

Hard disk with 4 Megabytes free.

One of the following graphics adapters:

- AT&T Model 6300 High-Res Mode (640x400)
- ATI Wonder Card (800x600)
- ATI Wonder Card 4M (1024x768)
- Genoa Super EGA (640x480)
- Genoa Super EGA Hi-Res (800x600)
- Genoa (1024x768)
- Hercules Monochrome Graphics Adapter
- HP Vectra (640x400)
- IBM Standard Color Graphics Adapter (CGA)
- IBM Enhanced Color Graphics Adapter (EGA)
- IBM Video Graphics Array (640x480)
- Multi-Color Graphics Array (640x480)
- NEC EGA MultiSync (640x480).
- Orchid ProDesigner (800x600)
- Orchid ProDesigner VGA Plus (1024x768)
- Paradise AutoSwitch EGA 480 (640x480)

Paradise Hi-Res. (800x600)
Paradise 1024 (1024x768)
Sigma Designs Color 400 (640x400)
SOTA VGA/16 (800x600)
SOTA VGA/16 (1024x768)
Tecmar Graphics Master (640x400 - Setup #1)
Tecmar VGA/AD (1024x768)
Toshiba T3100 (640x400)
Trident Impact (800x600)
Trident Impact (1024x768)
VEGA Video7 (720x540)
VEGA Video7 (800x600)
VEGA Video7 VRAM (1024x768)
Wyse Technology WY-700 (1280x800)

SCHEMA III supports a variety of mouses, dot matrix and laser printers, and pen plotters. Please refer to Appendix E, *Hardware Requirements*, for a list of these devices.

The READ.ME file, contained on SCHEMA III PROGRAM Disk 1, lists any newly-supported hardware and user features not covered in this manual. **Please read and/or print the READ.ME file.**

To view the READ.ME file:

ACTION:

At the DOS prompt in the \SCHEMA directory type: **SCHEMA <Enter>**. Once in the Program Manager under the **CONTROL** column select the command, **READ.ME** with a double mouse click or **<Enter>** twice.

RESULT:

The READ.ME file will be in the View utility of the Manager. Use the arrow or page up and down keys to

read the file.

To print the READ.ME file:

ACTION:

At the DOS prompt type:**COPY READ.ME PRN <Enter>**

RESULT:

This outputs the READ.ME file to your printer.

NOTE

All keyboard entries (and screen displays) appear as boldface text in this manual. Also, <Enter> refers to a single keypress of the <Return> key (the same is true for <Esc>, <Backspace>, etc.). Refer to Appendix C, Text Editing Keyboard Usage, for a complete listing of the editing keys used with SCHEMA III. If during the tutorial, your system does not respond as stated, refer to Appendix D, Error Messages.

1.3 DISK BACKUPS

Before proceeding, you should make backup copies of all the SCHEMA III disks. The disks are not copy protected and can be copied with the DISKCOPY command.

To create backup disk:

ACTION:

At the DOS prompt type: **DISKCOPY A: B: <Enter>**

RESULT:

This copies disk in Drive A: to Drive B:.

Refer to your DOS manual for specific instructions.

1.4 INSTALLATION

SCHEMA III is designed to be installed automatically. When SCHEMA III is installed, a hard disk system is required. During installation, the DOS CONFIG.SYS and AUTOEXEC.BAT files are automatically modified to facilitate SCHEMA III.

To install on Hard Disk System:

ACTION:

At the DOS prompt place Program 1 Disk into Drive A and type: **INSTALL <Enter>.**

RESULT:

Installation will begin.

*NOTE- During the installation, you will have an option of installing SCHEMA III in conjunction with PCB. If you do not own a copy of PCB, simply proceed with the default option and install SCHEMA III by itself.

1.5 TUTORIAL - SCHEMA III MADE SIMPLE

At this point, SCHEMA III should be installed on your system. SCHEMA III is very easy to use, *but there are fundamental procedures to learn before real productivity can be realized.* The following paragraphs provide a stepped tutorial to *quickly* introduce you to SCHEMA III's commonly-used programs and features. *Omation strongly recommends that you follow through this tutorial.* After doing so, refer to program

chapters for more information. Before proceeding, let's briefly review both the design and organizational functions of SCHEMA III.

These are the following Lessons:

LESSON 1 - AN OVERVIEW OF SCHDRAW
This tutorial will briefly familiarize the user to the commands of the Schematic Editor.

LESSON 2 - USING THE MANAGER AND SCHCFG
Demonstrate the use of the SCHEMA Program Manager and Configuration files.

LESSON 3 - USING SCHDRAW TO CREATE A SCHEMATIC
Describes the use of the Schematic Editor to place components and labels.

LESSON 4 - CREATE A PART USING THE OBJECT EDITOR
Introduce the user to the built in Object editing features and takes the user through the steps in creating one of their own.

LESSON 5 - FINISH THE LOGIC SCHEMATIC
Add the new component created to the schematic that was created earlier. This tutorial will demonstrate the editing of an existing schematic. Includes discussion of FILE commands, NOTES commands, EDIT commands, Regions , and the SEARCH commands.

LESSON 6 - USING HIERARCHY
Takes the user through the process of creating a hierarchical object and incorporating it into a hierarchical schematic. Also explains how to move though a hierarchical schematic using the PUSH\POP feature.

LESSON 7 - USING SCHPOST
The processing of the schematic is explained and the outputs that

SCHPOST generates are discussed.

LESSON 8 - USING SCHPLOT & SCHPRINT
Briefly show how to utilize the commands to output the schematic.

LESSON 9 - USING SCHLIB AND LIBRARY MANAGEMENT
Instructs the user on the library philosophy and management of library parts.

SCHEMA III Program Manager:

A multi-layered menu capable of organizing your entire hard disk. From this menu, you can invoke SCHEMA III's schematic functions or any other type of stored program or command.

SCHEMA III CAE functions:

SCHDRAW - The Drawing Editor uses a mouse or the keyboard to create schematics and general purpose drawings. This function is also used to create custom objects and symbols.

SCHPLOT/SCHPRINT - Printing and plotting programs.

SCHPOST - The Post Processor creates a bill of materials, Wire/Net/Pin Lists Usage Reports, DRCs, etc. from SCHEMA III schematics.

SCHCFG - Configuration File Manager creates and edits configuration files.
SCHLIB - Library maintenance utility manger creates new object libraries or edits existing libraries.

UTILITIES- Includes SCHFWD, SCHBACK, SCHLOCAL & SCHECO these provide such features as SCHEMA-PCB layout forward and backward annotation, virtual page creation, and command library summation.

Using a Mouse with SCHEMA III

Using a mouse with SCHEMA III improves basic editing speed and is highly recommended. In the program, each mouse button performs a different function. The left mouse button is used to select command boxes and perform drawing functions. If you have a three-button mouse, the middle mouse button can also be used to select commands. When the middle mouse button is pressed, the cursor's position is frozen in the Drawing Window and the Command Window Indicator is activated. The Command Window Indicator identifies the currently-selected command box and is moved by moving the mouse up or down. When the Command Window Indicator is positioned over the desired command, press the left mouse button to select the command. Next, the middle mouse button can be pressed to reactivate the Drawing Window cursor. The right mouse button is used by SCHEMA III to UNDO (<ESC>) a command.

IMPORTANT NOTE

Unless otherwise stated, all mouse references in this manual refer to the left mouse button.

Using the Keyboard with SCHEMA III

If you choose not to use a mouse, you have the option of using the keyboard. Some users prefer the keyboard to a mouse for drawing box shapes, lines, and placing text and graphics.

Command boxes are selected with the keyboard by first pressing the <F1> Function key. The <F1> key informs SCHEMA III that a

Primary Command is about to be selected. Next, press the **Capital** letter of the desired Primary Command Box. After the desired Primary Command is selected, the associated Secondary Command is executed by simply pressing the **Capital** letter of the desired Secondary Command Box. If the Text Window is displayed, any key press is interpreted as text. To override the Text Window, press the **<F2>** key and enter the **Capital** letter of the desired Secondary Command. All drawing functions are started and stopped using the Insert **<Ins>** key. If using the numeric pad, be sure that **Num Lock** is off. The **<Ins>** key operates like the left mouse button. Pressing **<Ins>** once is identical to pressing the left mouse button, while pressing **<Ins>** again is the same as releasing the left mouse button. Cursor movement is controlled with the Arrow keys (Up, Down, Left, and Right). To move the cursor right, simply press the Right Arrow key. To move the cursor up the drawing, press the **<Up Arrow>** key and so on. Chapter 3, *SCHDRAW*, and Appendix C, *Text Editing Keyboard Usage*, contain additional information on the keyboard and the editing keys used with SCHEMA III.

Auto Backup

In the event of a power failure or software hang you have a backup. SCHEMA backups up your schematic every five minutes. The file updated is called, BACKUP.P01. SCHEMA will use the OMATION environment variable to find the path to where to send the file. The interval between backups can be adjusted by using SCHCFG to change the SCHEMA.CFG *ONLY*. The variable is located in the Drawing Parameters Configuration. The numeric values are integer and in minutes. The default is five, but you can raise or lower this value. A value of **0** will disable this feature.

BEGINNING THE PROGRAM

Entering SCHEMA III is as simple as typing in the SCHEMA name.

ACTION:

At the DOS prompt type: **CD\SCHEMA <Enter>**

RESULT:

This enables you to operate from the proper directory.

ACTION:

Now type: **SCHEMA <Return>**

RESULT:

The SCHEMA III Program Manager will appear. You are ready to begin Lesson 1. ***Learning about the major functions of SCHEMA III will be far easier if you follow every step of Lesson 1. It is arranged in an Action/Result format that tells you every step to take and the result you should expect.***

LESSON 1 - AN OVERVIEW OF SCHDRAW

This lesson will guide you through that part of SCHEMA III you will be using most often, SCHDRAW. *If an area is not covered in this section of the tutorial, don't be concerned. Consult the on-line help or the index provided at back of this manual.* ***If you still can not resolve a question, call OMATION's support staff.***

ACTION:

Place the cursor on the word **SCHDRAW** in the second column of the command window and **double-click** the left mouse button.

RESULT:

A grid layout and a command window of SCHDRAW commands will appear. These commands are HELP, ZOOM, DRAW, EDIT, NOTES, FILE, CONFIG, and SEARCH. We will begin with the first:

SCHDRAW HELP

ACTION:

Click on the **primary command, HELP**, at the top left-hand corner of the screen.

RESULT:

This allows you to receive help on any one of the function keys, SCHDRAW commands and their corresponding secondary commands.

ACTION:

Click on the primary command **DRAW.**

RESULT:

The different secondary commands of DRAW are listed at the bottom of the command window.

ACTION:

Click on the **WIRE** secondary command of DRAW.

RESULT:

A Help screen offering guidelines on drawing wire will appear.

ACTION:

Click again on the word **HELP.**

RESULT:

The Help Mode is exited. You are now ready to go to the ZOOM section of this tutorial.

SCHDRAW ZOOM

ACTION:

Click on the word **ZOOM.**

RESULT:

The secondary commands of ZOOM appear below. They are 1:1, 2:1, 4:1, 8:1,VGA.

ACTION:

Click on the ratio **2:1.**

RESULT:

Viewing area is increased. Original screen is reduced to one-fourth of size.

ACTION:

If you are using a higher resolution graphics adapter you can click on VGA.

RESULT:

The graphics mode is switched to VGA. This can allow you to see more pixel definiton for creating parts placing text or lines.

ACTION:

Click on the ratio **1:1.**

RESULT:

The schematic is enlarged back to its original size.

ACTION:

Press the **<F8> key three times.**

RESULT:

The schematic performs a complete "zoom-out" cycle, finishing at the ratio 8:1.

ACTION:

Press the **<F7> key three times.**

RESULT:

The schema makes a complete "zoom-in" cycle returning to ratio of 1:1. You are ready to go to the DRAW section of this tutorial.

SCHDRAW - DRAW

ACTION:

Click on the word **DRAW.**

RESULT:

The various secondary commands of DRAW are listed in the lower half of the command window.

ACTION:

Click on the secondary command **OBJECT.**

RESULT:

A text window appears at the base of the screen allowing you to call up a specific family of objects.

ACTION:

Type **74LS*** in the text window and press **<Enter>.**

RESULT:

A list of all 74LS library objects in LIBR 1 appears on the screen. This library should be the TTL1.LBR.

ACTION:

Place the cursor on **74LS08** and press left mouse button.

RESULT:

Object 74LS08 appears on screen.

ACTION:

Click the **right** mouse button.

RESULT:

List 74LS* list reappears.

ACTION:

Press the **<Enter>** key.

RESULT:

The schematic editor reappears with a text window displaying "74LS08".

ACTION:

After 74LS08, **type in a comma and the reference designator U1.** Press the **<Enter>** key.

RESULT:

The window disappears and the mouse cursor reappears.

ACTION:

Place the cursor on the drawing, hold the left mouse button down and move the cursor across the screen to a desired position.

RESULT:

Object 74LS08,U1 moves with the cursor across the screen.

ACTION:

Release the left mouse button.

RESULT:

Object 74LS08,U1 is placed on the schematic.

**Note- After placing an object, SCHDRAW must be in the EDIT or DRAW mode before the object can be moved.*

ACTION:

Now click on **ROTATE**, in the lower half of the command window. This is another secondary command of DRAW.

RESULT:

You can now increment the object's 90 degree rotations and its DeMorgan equivalents.

ACTION:

Place the cursor on the object **74LS08 and click 8 times.**

RESULT:

The IC and its DeMorgan equivalents are rotated, returning to the original state.

ACTION:

Click on **GATE**, immediately beneath **ROTATE**, in the command window.

RESULT:

This allows you to toggle through the different gates of 74LS08.

ACTION:

Place the cursor on the object **74LS08 and click 4 times.**

RESULT:

The different gates appear on the circuit, returning to its original state of Pins 1,2 and 3.

ACTION:

Click on **WIRE**, the first secondary command of **DRAW**.

RESULT:

This allows you to draw wires.

ACTION:

Place the cursor on the drawing to the **left of Pin 1 of 74LS08** and while holding down the left mouse button, drag the mouse until a horizontal wire is drawn to the tip of Pin 1. Release the button. **Do the same for Pin 3,** on the opposite side of the circuit.

RESULT:

Wires are now connected to Pins 1 and 3.

ACTION:

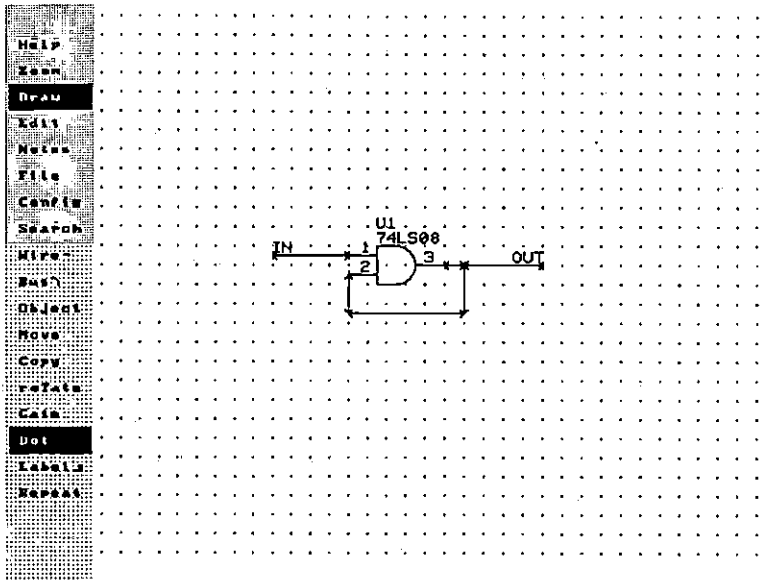
From the center of the wire connected to pin 3 **draw an L-shaped wire** (first moving the mouse down vertically,then horizontally beneath 74LS08) until the end of wire is parallel, to Pin 2.

RESULT:

This is the beginning of a feedback wire from Pin 3 to Pin 2.

ACTION:

Now **draw another wire** from the end of the wire just drawn **straight up to Pin 2.**



Finished SCHDRAW OVERVIEW schematic

RESULT:

The wire is completed for the feedback loop. The connection assuring electrical conductivity, however, is not complete until after the next step.

ACTION:

Place the cursor on DOT, another secondary command of DRAW and click. **Place the cursor where the feedback wire meets the wire connected to Pin 3 and click.**

RESULT:

The wire connected to Pin 3 is broken and a connection is made between the feedback wire and Pin 3. Placing a dot allows you to finalize this connection.

ACTION:

Place the cursor on LABEL, immediately beneath DOT and click. **Continue clicking** on the different label configurations.

RESULT:

These configurations represent the orientation of the label to the definition header. The label itself is represented by the square, and the wire is represented by the line. This allows you to Label inputs and outputs of the circuit accordingly. A text box will now appear at the bottom of the screen.

ACTION:

Click the mouse button on LABEL until appears to the right of the word command LABEL. Now type **the word, "IN" into the text box and press <Enter>.**

RESULT:

The text box disappears.

ACTION:

While holding the left mouse button down, **drag the cursor to the point where "IN" sits above the wire to Pin 1 and the vertical portion of the header just meets the end of the wire.** Release the button.

RESULT:

The wire has been electronically "broken" and the IN Pin of the circuit has been labeled.

ACTION:

Clicking on **LABEL** again, find the configuration. **Type "OUT" into the text box and press <Enter>. Place the "OUT" label at the end of the wire attached to Pin 3.**

RESULT:

The labeling is complete and you may continue to the EDIT section of this tutorial.

SCHDRAW EDIT

ACTION:

Place the cursor on the primary command, **EDIT** of the SCHDRAW command window and click.

RESULT:

The secondary commands of **EDIT** will appear in the lower half of the command window.

ACTION:

Place the cursor on the secondary command, **MOVE**, and click. **Place the cursor on the object U1, hold the left mouse button down and drag it to another part of the screen.**

RESULT:

The object will move with the cursor, away from the wires connecting to it.

ACTION:

Press the **right** mouse button to **<Esc>** or **UNDO**.

RESULT:

The object goes back to its original position.

ACTION:

Place the cursor on the **ERASE** secondary command of EDIT and click. **Place the cursor on the "OUT" label by the wire connected to pin three and click again.**

RESULT:

The label is erased.

ACTION:

Press the **right** mouse button, **<Esc>** or **UNDO**.

RESULT:

The label "OUT" reappears.

ACTION:

Click on the IMAGE secondary command of EDIT. **Place the cursor on U1 and click again.**

RESULT:

A text window appears displaying "74LS08,U1"

ACTION:

Backspace over the reference designator, U1, and type: U2 <Enter>.

RESULT:

The circuits reference designator is changed to U2.

ACTION:

Click on the WIRE secondary command of EDIT. Place the cursor on the end of a wire or a wire juction.

RESULT:

The wire will move. Try again at the other end of a wire. This will break the wire and allow the user to alter the wire path. Note: EDIT-WIRE also allows the modification of NOTES-LINE as well. You may now move on to the NOTES section of this tutorial.

SCHDRAW NOTES

ACTION:

Place the cursor on the primary command NOTES in the SCHDRAW command window and click.

RESULT:

The secondary commands LINE, CIRCLE and TEXT M will appear in the lower half of the command window. All notes constructed using this primary command are non-electrical.

ACTION:

Click on LINE and continue clicking observing the icons to the immediate right.

RESULT:

These icons correspond to the different types of line (not electrical wire) you can create. Lines can be thin, thick, dotted, or dashed.

ACTION:

Click to the dotted line, place the cursor on the drawing, click the left mouse button, and drag the mouse until you are familiar with drawing a line.

RESULT:

A dotted line is created.

ACTION:

Next, click on the secondary command CIRCLE.

RESULT:

A text box will appear at the bottom of the screen.

ACTION:

Type in the number 40 - the radius of a circle. This radius is measured in pixels and each visible grid is separated by 16 pixels. Press <Enter>.

RESULT:

The specifications of a circle with a radius of 40 has been entered.

ACTION:

Place the cursor on the drawing, hold down the left mouse button and **drag the cursor to the desired position. Release the mouse button.**

RESULT:

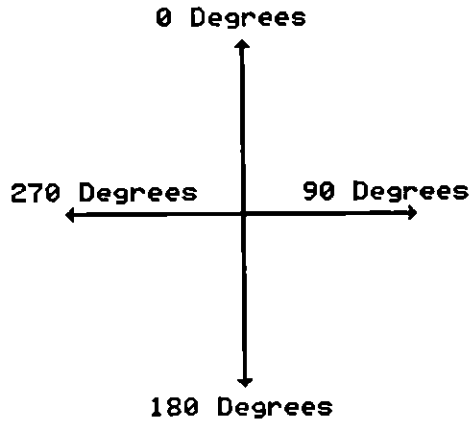
The specified circle follows the cursor and is placed in the desired spot. (Remember, the right mouse button will UNDO any undesired creation)

ACTION:

Again select circle. In the text box type 12,5,90. Press <Enter>.

RESULT:

This is the means of drawing an arc, (a portion of a circle). The first number specifies the radius, the second specifies the beginning degree and the third specifies the ending degree.



ACTION:

Place the cursor on the drawing, hold the left mouse button down, **drag to a desired position, and release.**

RESULT:

The specified arc follows the cursor and is placed in the desired spot.

ACTION:

Place the cursor on the TEXT secondary command and click. **Continue clicking through the choices S,M, and** L. These stand for small, medium, and large. **Stop on L and type your name in the text box at the bottom the screen. Press <Enter>.**

RESULT:

You have selected large type for your name.

ACTION:

Place the cursor on the drawing, hold down the left mouse button and **drag the cursor to any desired position. Release the button.**

RESULT:

Your name appears in large type and follows the cursor to the desired position. You may now go to the **FILE** section of this tutorial.

SCHDRAW FILE

ACTION:

Place the cursor on the primary command, **FILE**, of SCHDRAW command window and click.

RESULT:

The **FILE** secondary commands appear in the lower half of the command window.

ACTION:

Click on the **SAVE** secondary command.

RESULT:

A text window will appear reading "NONAME.P01".

ACTION:

Use the left arrow key to move back and replace "NONAME" with the name "OVERVIEW". Leave the .P01 as it is. Now press <Enter>.

RESULT:

The drawing will be saved under this new name.

ACTION:

Place the cursor on the secondary command **LOAD** and click. **Again, erase "OVERVIEW" and place an * in front of the .P01. Press <Enter>.**

RESULT:

The * is a wildcard that will allow you to browse all .P01 files and select the one you want. If you had wanted to, however, you could have simply typed in the desired file you wanted to load.

ACTION:

Highlight **OVERVIEW.P01, press <Enter>** or click the left mouse button.

RESULT:

A text window will appear displaying the file you have selected for loading.

ACTION:

Press **<Enter>.**

RESULT:

The file you have just saved loads onto the screen. You may now go to the **CONFIG** (configuration) section of this tutorial.

*Note- **FILE-REDRAW** allows you to sharpen and repair a drawing after portions are inadvertently erased. This occurs when one section is erased or moved while overlapping another section. To repair simply click on **REDRAW.**

SCHDRAW CONFIG

ACTION:

Place the cursor on the primary command, **CONFIG** of the SCHDRAW command window and click.

RESULT:

CONFIG secondary commands appear in the lower half of the command window. The commands highlighted are default settings which are usually turned on. These consist of **V Grid, C Grid** and **XY Wdw.** Each of these will be explained.

ACTION:

Click on XY Wdw. Click again.

RESULT:

This turns off and on the window displaying the cursor's vertical and horizontal coordinates.

ACTION:

Select the **LIBR 1** secondary command of CONFIG. A text box containing **TTL1.LBR** will appear. This is your present library. **Continue clicking** through all of the libraries until you come back to this one.

RESULT:

You have reviewed the object libraries available to you. If you wish to alter which libraries are used, type the change and press **<Enter>**. If you wish to browse the libraries available to you , type a path with ***.LBR** and use the cursor to select the library.

ACTION:

Place the cursor on the secondary command **V Grid and click. Click again.**

RESULT:

All grid points on the drawing disappear, then reappear. **V Grid** means "Visual Grid".

ACTION:

Click on C Grid.

RESULT:

This turns off the snap-to cursor grid allowing you to move the cursor at one pixel increments.

ACTION:

Again, click on the **C Grid.**

RESULT:

This turns on the snap-to cursor. It now moves in eight pixel increments, forcing drawing commands to follow the grid patterns. Continue to the **SEARCH** section of this tutorial.

SCHDRAW SEARCH

ACTION:

Click on the **SEARCH** command of SCHDRAW.

RESULT:

A list of **SEARCH** secondary commands appear in the lower half of the command window.

ACTION:

Click on the secondary command **LABEL**, and a text box will appear. **Type: * (wildcard) and press <Enter>.**

RESULT:

The first label which you placed on the circuit will appear in the text box (IN).

ACTION:

Click the mouse button again.

RESULT:

The next label which you placed will appear in the text box (OUT).

ACTION:

Backspace over this label in the text box and change to "CLK". Press <Enter>.

RESULT:

The label is changed. If you click the mouse button again, a message will be displayed, "NO MORE MATCHES". You may now continue to the CONCLUSION of this lesson.

CONCLUSION

ACTION:

Select the primary command FILE from the SCHDRAW command window and select secondary command SAVE to record the changes. Either change the name and press <Enter> to keep both the new and old version of this drawing, or simply press <Enter> to save over the old drawing.

RESULT:

All changes are retained into memory and recorded to disk. A backup file with the same prefix and .Bnn extension is created (nn= the two trailing digits representing the page number, 01).

ACTION:

Select secondary command **QUIT** and press **<Enter>**.

RESULT:

SCHDRAW will be exited and you will Enter to the Program Manager. **It's time to take a break; click twice on "Quit" in the first column of the manager. You will be returned to DOS.**

For more detailed information on SCHDRAW, continue with the advanced tutorials in this chapter and see Chapter 3.

LESSON 2 - USING THE MANAGER AND SCHCFG

1.5.2 The SCHEMA Program Manager.

The SCHEMA Program Manager (hereinafter referred to simply as 'the Manager') is used to run SCHEMA III's programs.

The Manager is user definable. The user can alter the commands by invoking the **EDIT** command in the **CONTROL** column. This takes the user into the Edit-Mode. The user can then select an existing command or a blank command and define how to invoke it.

The Manager has online Help. If you select the **HELP** command from the **CONTROL** column, the Help-Mode will be invoked. Then select any of the SCHEMA commands and a help file will be brought up for you to view. **EDIT** and **HELP** are discussed in more detail in Chapter 2.

The Manager, now displayed on your screen and shown in figure 1-1 or 1-1A, lists SCHEMA III programs.

To start a new schematic, we will first select a blank drawing page. Select **NEW_PAGE** using the mouse or the **<Arrow>** keys to highlight the command and press the mouse button or **<Enter>** twice. **NEW_PAGE** changes the **FILTER** to ***.P99**, uses the **PATH** where the executable and blank drawings are stored and automatically invokes the command, **DRAWINGS**. The available blank drawing pages (files ending with **.P99**) are displayed on the Manager Menu.

For the tutorial exercise, we will select file **BPAGE2.P99** which is a virtual D-size blank page with a predrawn border and title block area (Chapter 3, *SCHDRAW*, contains a complete discussion of SCHEMA III blank pages.).

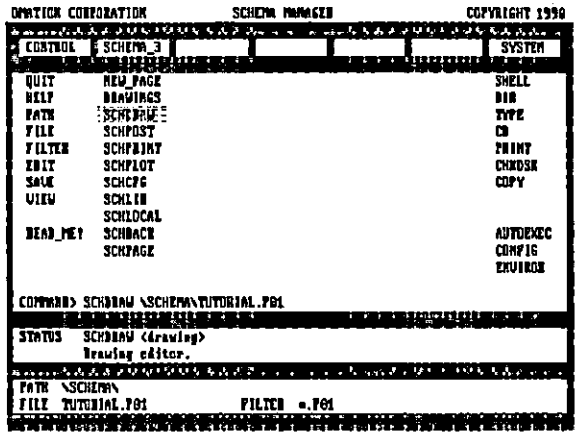


Figure 1-1. The SCHEMA Program Manager

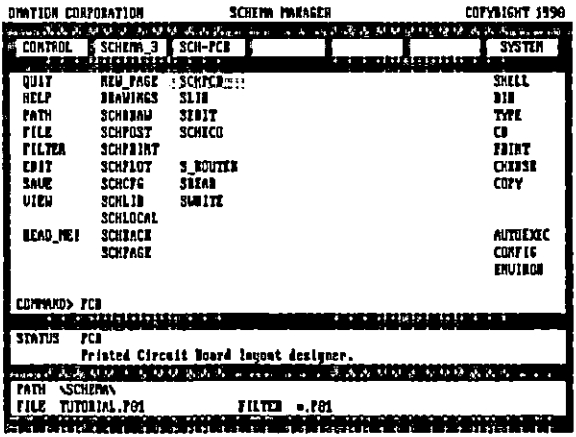


Figure 1-1A. The SCHEMA-PCB Program Manager

To select **BPAGE2.P99**, use the mouse or the **<Arrow>** keys to position the cursor over **BPAGE2.P99**. **BPAGE2.P99** lights when the cursor is positioned correctly. Next, press **<Enter>**. The Manager prompt line displays:

BLANK PAGE SELECTED, CREATE A NEW DRAWING PAGE?
Type yes or no (Y or N), ESC to exit

Press **Y** to create a new page. The following prompt appears:

TYPE IN NEW PAGE NAME

The cursor flashes at the bottom of the Manager menu to let you know that SCHEMA III is expecting an entry. SCHEMA III schematics are always named as follows:

DRAWING.P01 - First page of schematic
DRAWING.P02 - Second page of schematic
* * *
* * *
DRAWING.P98 - 98th page of schematic (maximum)

where **DRAWING** is the generic drawing name. The **PATH** which is where the Executable files and blank pages are stored (typically \SCHEMA) will be displayed. If this is not the correct path, you may use the **<Backspace>** key to delete it and enter the correct path/filename.extension (**DRAWING.Pxx**). We will name our schematic **LOGIC**. Since this is the first page of the LOGIC drawing, enter:

LOGIC.P01 <Enter>

Keep in mind that the total length of the drawing name (excluding extension) is limited by DOS to no more than 8 characters. After **<Enter>** is pressed, the following is displayed in the Manager prompt

line:

CREATE A NEW CONFIGURATION FILE?
Type yes or no (Y or N), ESC to exit

When a new drawing is started, a configuration file is normally created to assign the drawing's title block data and identify other configuration items unique to the drawing. Since we are creating a new drawing, we need to create the drawing's configuration file, so press **<Y>**. When **<Y>** is pressed, a configuration file is made in the current directory and named as LOGIC.CFG to match our new drawing.

The Manager menu is replaced with the SCHCFG Main Configuration Menu, shown in figure 1-2. Before SCHDRAW is used to actually draw our schematic, we will edit the LOGIC configuration file to define the drawing's title block information.

1.5.3 SCHCFG

SCHCFG is used to create and edit SCHEMA III configuration files. There are two types of SCHEMA III configuration files: SCHEMA.CFG and DRAWING.CFG (where DRAWING is the name of the associated schematic drawing - in our case: LOGIC).

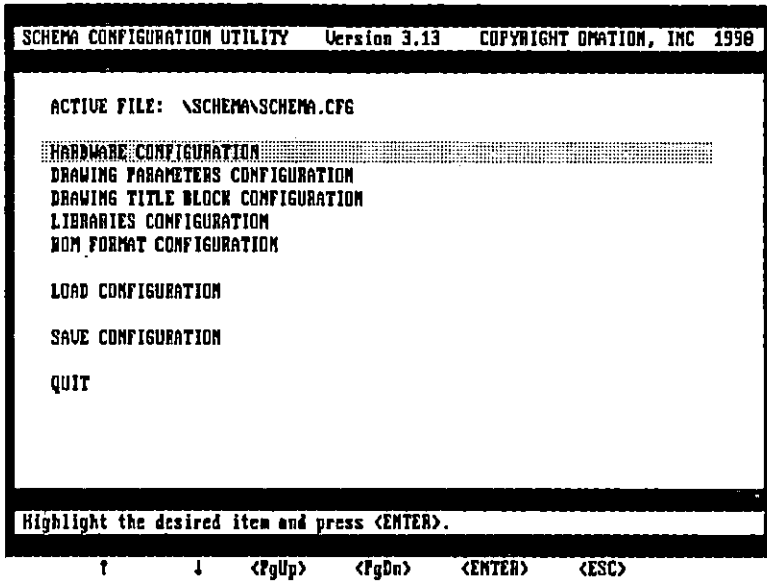


Figure 1-2. SCHCFG Main Configuration Menu

When SCHEMA III was installed on your system, a SCHEMA.CFG file was created. The SCHEMA.CFG file is a template file used to create your DRAWING.CFG files. SCHEMA III programs first look for the SCHEMA.CFG in the current directory. If not there, then SCHEMA III looks in the directory specified by the OMATION environment variable. The only parameters that it retrieves are the graphics and printer driver variables (GRAPH= and PRINT=). Then SCHEMA III searches for DRAWING.CFG in the same directory as the drawing to get all of the remaining variables. If no DRAWING.CFG file is found then the rest of the variables are taken from the SCHEMA.CFG file.

DRAWING.CFG files contain information unique to the associated

drawing. DRAWING.CFG files contain the associated drawing's title block information, the drawing's BOM output format, and the type and number of object libraries assigned to the drawing.

For now we will use SCHCFG to enter just the LOGIC schematic's title block information in the LOGIC configuration file. Chapter 5, *SCHCFG*, discusses SCHCFG in detail.

Currently SCHCFG is displaying the Main Configuration Menu, displayed in figure 1-2. Use the mouse or **<Arrow>** keys to highlight **DRAWING TITLE BLOCK CONFIGURATION.** Next, press the mouse button or **<Enter>.**

The Drawing Title Block Configuration Menu consists of eight variables used to identify the drawing name, your company's drawing number assigned to the drawing, the drawing revision number, the date, the maximum page number of the drawing, and so on. The variables are selected by simply highlighting the variable and pressing **<Enter>.** Variables **F6, F7,** and **F8** are undefined and can be used to carry any additional required information. Pointers to bring these variables into the title block of a page can be created using **NOTES-TEXT** see Chapter 3.

When the SCHDRAW program reads the LOGIC configuration file, it copies the text assigned to the title block variables and places it in predetermined locations within the LOGIC.P01 drawing page title block. When additional schematic pages are created, SCHDRAW automatically fills the title blocks of each page with the preassigned information. The idea behind this process is to assign a drawing's title block information once, so every time a new drawing page is started, the title block information does not have to be entered again. The end result is improved consistency between schematic pages with a minimum number of keystrokes.

Text assigned to the Drawing Title Block Configuration Menu variables is also printed across the top of SCHEMA III SCHPOST output reports. This conveniently associates SCHPOST outputs with a given schematic.

Now let's define the LOGIC drawing title block contents. Select **F1** and press **<Enter>** to enter the drawing name to be displayed in the LOGIC title block. When **<Enter>** is pressed, the following prompt is displayed:

Editing configuration parameter F1

Press **<Esc>** to clear the current entry and enter:

LOGIC SCHEMATIC <Enter>

After **<Enter>** is pressed, the entry is displayed in the upper left corner by the **F1=** variable. Next, select **F2** and press **<Enter>** followed by **<Esc>** and enter the following sample drawing number:

100-001-0001D <Enter>

Now select **F3** and press **<Enter>** followed by **<Esc>** and enter the following sample revision letter:

1A <Enter>

Select **F4** and press **<Enter>** followed by **<Esc>** and enter today's date in any format, followed by **<Enter>**. Finally, select **F5** and press **<Enter>** followed by **<Esc>** and enter the maximum page number:

<Enter>

This completes our LOGIC title block assignments.

Now select **EXIT** and press **<Enter>**. The **LIBRARIES CONFIGURATION** command is highlighted. Object libraries are assigned to the applicable drawing from this menu. When drawing schematics, SCHDRAW searches for objects in the order libraries are assigned in this menu. **LIBR1** is searched first, followed by **LIBR2** and so on.

NOTE

When you are finished with the tutorial,, you may want to change the LIBR1 = statement in your configuration file. The default library is LIBR1. After completing the tutorial, you may want to create a library using SCHLIB and assign it as LIBR1. Then, the objects you create can be contained in this library. We will attempt this later in the manual.

Select **EXIT** and press **<Enter>** or the mouse button and the **BOM FORMAT CONFIGURATION** command is highlighted. Then, press **<Enter>**. This menu is used to specify the drawing's Bill of Materials (BOM) output format. After the LOGIC schematic Bill of Materials is printed later in the tutorial, you may discover that you would like to change the BOM format to match your company's format. This menu lets you change the BOM format to suit your needs.

Select **EXIT** and press **<Enter>**. We have returned to the SCHCFG Main Configuration Menu. Save the LOGIC configuration file by selecting **SAVE CONFIGURATION** and pressing **<Enter>**. The current configuration file name is displayed in the text line. Press **<Enter>**. The LOGIC configuration file is saved to the current directory (typically \SCHEMA). We will now exit SCHCFG and return to the Manager to begin drawing the LOGIC schematic. Select **QUIT** followed by **<Enter>**. The SCHCFG Main Configuration Menu is replaced with the Manager.

When the Manager returns the current **FILE** is **LOGIC.P01.**

LESSON 3- USING SCHDRAW TO CREATE A SCHEMATIC

1.5.4 SCHDRAW

Be sure that you completed Lesson 2 and created the blank page LOGIC.P01. To invoke SCHDRAW with the LOGIC.P01 blank page, highlight the **DRAWINGS** command and select LOGIC.P01. Then select the **SCHDRAW** command and press **<Enter>** or the mouse button. After SCHDRAW loads, the screen displays the Schematic Editor with the upper left corner of the LOGIC.P01 drawing, as shown figure 1-3.

The SCHDRAW screen is divided into a primary and secondary command windows. Along the left side of the screen is the Command Window. The Command Window is comprised of individual Command Boxes. All SCHDRAW commands appear in Command Boxes. When a command is selected (enabled), the applicable Command Box lights. Commands are selected using the keyboard or a mouse. The commands currently shown in the Command Boxes are Schematic Editor Primary Commands (**DRAW, EDIT, CONFIG,** etc.). Some of these commands have additional, subordinate commands called Secondary Commands. Secondary Commands appear in the lower portion of the Command Window when the associated Primary Command is activated.

The main screen area is referred to as the Drawing Window. The Drawing Window displays the current drawing (when in the Schematic Editor) or the current object (when in the Object Editor). In either editor, the Drawing Window displays a portion of the total drawing or object. Panning allows other portions of the drawing/object to be seen. The **ZOOM** command shows different viewing perspectives of the drawing/object.

Another visible item on the screen is the cursor. The cursor is

represented by a small arrowhead. The cursor is used to draw and select commands and is controlled by the keyboard or a mouse.

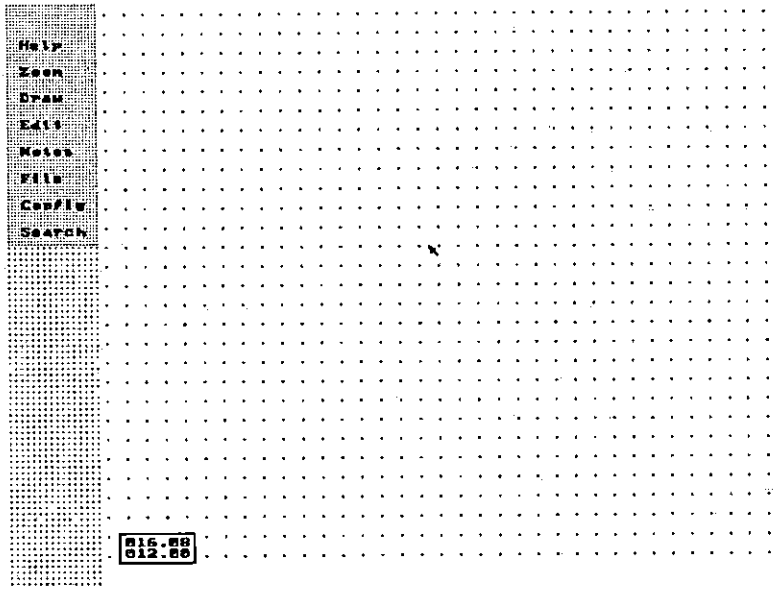


Figure 1-3. SCHDRAW Schematic Editor

When some commands are selected, the cursor may disappear for a moment. When this happens, SCHDRAW is performing an operation. After the operation is complete, the cursor reappears.

The small box in the lower left corner of the Drawing Window is the Cursor Coordinate Window. Think of the Cursor Coordinate Window as an electronic ruler. The two numbers in this window represent the current cursor location within the drawing. The upper number is the cursor's X-location and the lower number its Y-location, relative to X/Y

position 0,0 which is at the extreme upper left corner of the drawing. Two other windows appear only under special circumstances. These windows are shown in figure 1-4.

The Text Window appears any time keyboard entry is required. Characters are displayed in the Text Window as they are entered from the keyboard.

The Error Window appears only when an error occurs and contains an error message. When the error window appears, you must press <ESC> on the keyboard or the right mouse button to acknowledge the error and remove the error window. Error messages are summarized in Appendix D.

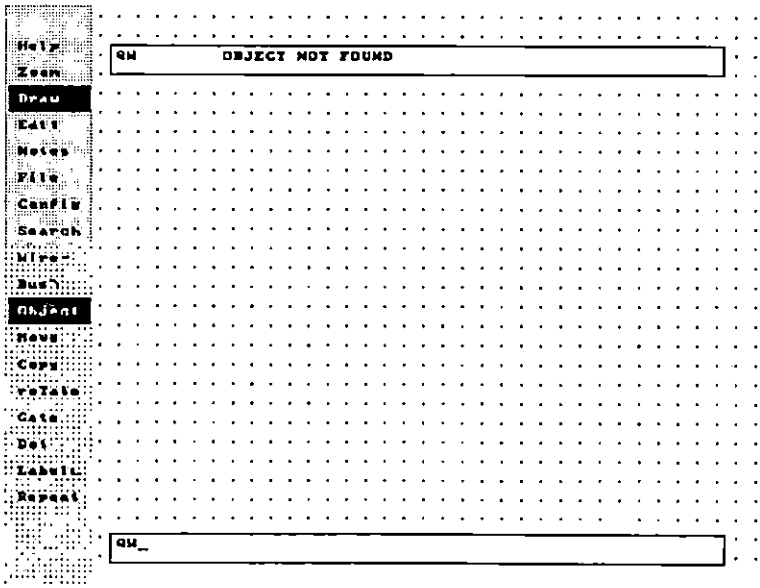


Figure 1-4. Text and Error Windows

1.5.4.3 The Schematic Editor

We are presently in SCHDRAW's Schematic Editor which is used for creating/editing drawings. You can not use the Schematic Editor to manipulate or edit objects - the Object Editor, described later, is used exclusively to create and edit objects and symbols. When SCHDRAW is invoked, it always default-loads the Schematic Editor. Let's begin the Schematic Editor tutorial with panning techniques.

To pan the drawing page, move the cursor to the right edge of the Drawing Window. Once the cursor touches the edge of the Drawing Window, you will notice that the drawing begins to pan to the right. Now move the cursor to the bottom edge of the Drawing Window and watch the Drawing Window as it pans down the drawing. Moving the cursor to any screen border causes the Drawing Window to pan the drawing in the appropriate direction.

Select **HELP**. The **HELP** Command Box lights and a listing of all the keyboard keys recognized by SCHDRAW is displayed in the Drawing Window. Press any key to remove the listing.

Keyboard users can instantly move the cursor to the center of the Drawing Window or 'jump' the cursor to any screen border.

To instantly move the cursor to the center of the Drawing Window, press the **<Home>** key.

To jump the cursor to the left or right of the screen border, simultaneously press the **<Ctrl>** and applicable **<Arrow>** key. For example, **<Ctrl><Right Arrow>** jumps the cursor from its current position to the right edge of the Drawing Window causing the drawing to pan right.

Press the **<PgDn>** key to jump the cursor to the bottom edge of the Drawing Window and pan the drawing down. Press the **<PgUp>** key to jump the cursor to the top edge of the Drawing Window. To stop panning, simply press the opposite **<Arrow>** key.

Select **ZOOM**. The **ZOOM** Command Box lights and a set of Secondary Commands appears in the lower portion of the Command Window with the 1:1 perspective box also lit. You can pan the drawing by moving the cursor toward any screen border. Actual drawing functions can be done from either the normal or **ZOOM** views.

The five Secondary Commands available in the Schematic Editor are **1:1, 2:1, 4:1, 8:1** and **VGA** viewing perspectives. The **1:1** viewing perspective is the 'normal' view. The area displayed will be two times larger both horizontally and vertically if the 2:1 option is chosen. Likewise, the area displayed will be four or eight times larger if the **4:1** or **8:1** option is chosen.Hot keys for Zoom are: **<F7>** Zoom IN and **<F8>** Zoom OUT. The **VGA** command is for those who are using higher resolution adapters and would like more pixel resolution for creating objects or placing text and lines.

Select **2:1**. The **2:1** box lights and a dashed, rectangular box appears in the Drawing Window. (If the box does not appear, pan the drawing until it is visible.). This dashed box represents the area of the drawing displayed in the Drawing Window when **1:1** is selected (normal drawing view). Select **4:1**, notice the size of the drawing as you change the perspective from **2:1**. Position the cursor in the Drawing Window and either press the **<Ins>** key or press and hold the mouse button. The upper left corner of the box instantly moves to the cursor's position. As long as the mouse button is pressed, you can move the dashed box anywhere over the drawing. The Drawing Window pans if the box is moved to any border.

When changing from the different viewing perspectives, *the dashed box will automatically appear in the same position on the drawing as the previous screen.*

Releasing the mouse button (or repressing the **<Ins>** key) fixes the box's location. Position the box over the drawing's title block, located in the lower right corner of the drawing. Now select the 1:1 box. Now in normal viewing mode, the Drawing Window displays the title block area that was inside the box. Notice that the title block data we assigned to the LOGIC configuration file (using the SCHCFG program) is displayed in the title block. Experiment with **ZOOM** until you are comfortable with its operation.

Ensure the Drawing Window is displaying the normal view of the drawing (1:1 box lit). Now select **DRAW**. The **DRAW** Command Box lights and a set of subcommands (Secondary Commands) appears in the lower Command Window. These **DRAW** Secondary Commands are used to add images to a schematic drawing.

WIRE is used to draw signal and bus wires, **BUS** is used to add curved right-angle bends to signal and bus wires, **OBJECT** is used to retrieve an object (IC, resistor, diode, etc.) from one of the SCHEMA III object libraries, **DOT** is used to symbolize a connection, **LABEL** is used to add labels (identifier text such as signal names) to signal wires, and **REPEAT** is used to quickly repeat the last drawing command.

We will now draw the tutorial schematic. Figure 1-5 shows the LOGIC schematic. The following paragraphs provide a step-by-step procedure to draw the schematic.

Add LS Components and Assign Reference Designators
Select **OBJECT**. The **OBJECT** Command Box lights and the Text Window appears. Enter: **74LS00 <Enter>**

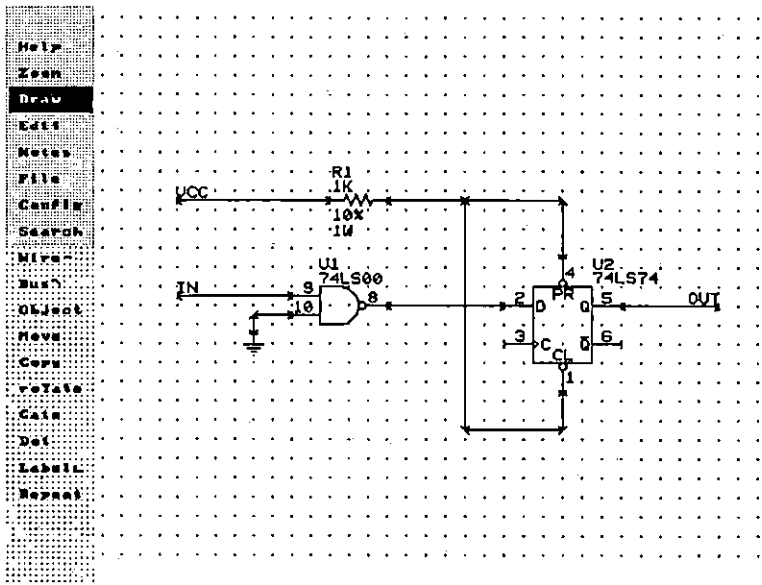


Figure 1-5. Tutorial LOGIC Schematic

If you are using a mouse, simply press and hold the left mouse button to position the 74LS00. Release the mouse button to place the 74LS00.

If you are using the keyboard to control the cursor, press **<Ins>** and use the **<Arrow>** keys to center the 74LS00 component in the Drawing Window. Next, press **<Ins>** again to place the component.

Now let's add another component to the drawing. Enter:

74LS74,U2 <Enter>

and place the 74LS74 to the right of the 74LS00, as shown in figure 1-5. Notice that a reference designator (e.g. **.AAB01**) was automatically assigned to the 74LS00 by the Schematic Editor when it was called up, while we preassigned reference designator **U2** to the 74LS74. We can easily change the 74LS00 reference designator. Select **EDIT** followed by **IMAGE**. Next, position the cursor over the 74LS00. Now either press **<Ins>** twice or click the mouse button. The Text Window appears with the 74LS00 **DRAW-OBJECT** *Parameter String*. **<Backspace>** over **.AAB01** and add a comma followed by **U1** after the 74LS00 entry. The Text Window displays:

74LS00,U1

After **<Enter>** is pressed, the **.AAB01** reference designator is instantly changed to **U1**. We can even change the gate section of the 74LS00, if desired. Select **EDIT-GATE** twice and the C gate is displayed.

The other method would be use **EDIT-IMAGE** again by positioning cursor over the 74LS00 and either press **<Ins>** twice or click the mouse button. Use the **<Backspace>** key and enter:

74LS00\1C,U1 <Enter>
The 74LS00 instantly changes to the C-gate section, or select **DRAW-GATE** twice.DeMorgan equivalents are also available for all the gates supplied with SCHEMA III.

The entry: **74LS00\2B,U1** retrieves the B-gate DeMorgan equivalent, while: **74LS00\2D,U1** retrieves the DeMorgan D-gate and so on.

Rotated versions of all gated objects are supplied with each library. You may increment through the available rotated items and DeMorgan equivalents by selecting the **DRAW-ROTATE** or **EDIT-ROTATE** and repeatedly toggling on the component.

The other method would be to change it on the **DRAW-OBJECT** Command line. You can now select the rotated orientation for any of the resident gates. For example:

74LS00\3C <Enter>

retrieves the 74LS00 C-gate in the bottom-to-top orientation.

74LS00\5D <Enter>

retrieves the 74LS00 D-gate in the right-to-left orientation.

Leave the standard 74LS00 C-gate in the Drawing Window.

NOTE *The ROTATE and GATE commands are convenience features that quickly increment through available rotated or gated versions of a part. These commands are quicker than using EDIT-IMAGE to alter the parts.*

1.5.4.3.1 BROWSING the Libraries

Now select **DRAW-OBJECT** again. String searches can be also be done from **DRAW-OBJECT**. For example, to see a listing of all 74LS devices in the drawing's associated object libraries, enter:

74LS* <Enter>

After **<Enter>** is pressed, a listing of all 74LS devices in SCHEMA III object libraries is displayed in alphanumberic order in the Drawing Window. Note that the DOS wildcard character (*) is supported to denote string searches. Entering **n=***, where **n** is the library number, will display all objects in that library. *The default is **n=1**.*

Use the arrow keys to highlight the desired object. Press the **<Ins>** key to view the object. If the object does not fit on one screen, use only the arrow keys to scroll the object. Press the **<Esc>** or **<Ins>** key to return to the Directory Screen. Then press **<Esc>** to return to the original drawing in the Drawing Window with the original library displayed in the Text Window.

Press **<Enter>** to add the selected object to the original drawing. After a slight pause, the drawing will appear with the object name displayed in the Text Window. Press **<Enter>** and then place the object with the left mouse button or **<Ins>** key. At this point the parameters can be defined the same way as when normally using the **DRAW-OBJECT** command. Refer to the **DRAW-OBJECT** command in Chapter 3, *SCHDRAW.*

Connect Components and CONFIG

Now let's connect the two LS components we placed in the Drawing Window. First, select **WIRE** and position the cursor at the exact ending point (pin stub bar) of the 74LS00, pin 8. Then either press **<Ins>** or the mouse button to begin a thin signal wire from pin 8 to the 74LS74, pin 2. When the wire touches the pin stub bar of the 74LS74, pin 2, either press **<Ins>** or release the mouse button. Refer to figure 1-5. Notice that small arrowheads appeared at the beginning and end points of the signal wire as it was drawn. These small arrowhead-type symbols are called *Wire Headers.* Wire Headers greatly aid signal wire placement by graphically illustrating the beginning and end points of signal wires.

Select **CONFIG** followed by **WI HDR.** The Wire Headers are removed. Reselect **WI HDR** to turn the Wire Headers back on. Be sure to always keep Wire Headers on when drawing schematics. Other **CONFIG** Secondary Commands are: **RPT, LIBR, PAN, V GRID, C GRID, DIAG, XY WDW, COLORS, and SAVE.**

The RPT Secondary Command determines the X and Y offset values (in pixels) that images are repeated from the last image when the DRAW-REPEAT command is used.

The LIBR Secondary Command identifies the path and name of the current configuration file object library assignments. Library assignments can not be changed from LIBR - they can only be viewed. Library assignments are made/changed using the SCHCFG program, discussed in Chapter 5.

The PAN Secondary Command is used to adjust the Drawing Window panning speed.

Throughout the tutorial, the Drawing Window has been covered with a dotted grid. This is referred to as the *Visual Grid*. The Visual Grid (V GRID) greatly aids drawing and, generally, should be activated when using SCHDRAW. The *Cursor Coordinate Window* displays the cursor's position within the drawing. The Cursor Coordinate Window is located in lower left corner of Drawing Window. The Cursor Coordinate Window displays X and Y coordinates based on whole visual grid (V GRID) units (2.00, 4.00, 33.00, etc.) and 0-15 pixel fractions (2.08, 4.11, 33.07, etc.) beginning at location 0.00/0.00 in the upper left corner of the drawing. The Cursor Coordinate Window is useful when creating objects (can be used as an electronic ruler) and required when locating errors identified by the SCHPOST post-processors. Thus, the Cursor Coordinate Window is normally displayed. The window is displayed (or hidden) by setting a configuration variable, XY=, On or Off in the drawing's configuration file. Pressing the **<F5>** key will toggle it on or off. The drawing's configuration file (LOGIC.CFG) contains SCHDRAW initialization CONFIG values. When these CONFIG commands are changed in SCHDRAW, the configuration file is unaffected. Configuration file initialization values can only be changed by using the SCHCFG program to edit the drawing's configuration file.

The *Cursor Grid* (**C GRID**) snaps cursor movement to an invisible 8-pixel grid and *must be activated when drawing schematics* to ensure exact "component pin header to wire" alignment. Each cursor grid unit is equal to one-half of a visual grid unit, i.e., you can position the cursor on or half-way between the visual grid dots when **C GRID** is selected. **C GRID** may also be toggled on and off by pressing the **<F10>** key.

DIAG is used to draw diagonal lines and is discussed later.

XY WDW is used to hide or display the cursor coordinate window.

COLORS allows you to alter the SCHDRAW COLORS online.

SAVE allows you to save the configuration back to DRAWING.CFG

Assign Signal Labels and Component Parameters

Select **DRAW** followed by **WIRE**. Now, route a signal wire from U1, pin 9, left about five visual grid units. Let's label this wire. The **LABEL** Secondary Command is always used to assign names to signal (thin) wires. Select **LABEL**. The **LABEL** Command Box lights and the Text Window appears in the Drawing Window. An icon is displayed next to **LABEL**. There are eight label types available, each providing a different physical label orientation. The desired label type (orientation) is obtained by selecting the **LABEL** Command Box. The icon in the command box shows the currently-selected label type. The small square dot portion of the command box icon represents the label while the line portion represents the wire relationship with respect to the label. Currently the icon should display the small square box above the line and close to **LABEL**. Upper or lowercase is available when using the **LABEL** command. Enter:

IN <Enter>

The Text Window disappears. Position the cursor in the Drawing Window and press the mouse button or **<Ins>**. The label IN appears and can be moved throughout the drawing as long as the mouse button is pressed. Position the label until it just touches the wire header at the left end of the wire connected to U1, pin 9. When the label is positioned correctly (figure 1-5), either release the mouse button or press **<Ins>** to place the label. Labels can be aligned vertically by using the vertical icons, if desired - refer to the **DRAW-LABEL** command description contained in paragraph 3.5 of Chapter 3 for additional information.

We will now assign ground to pin 10 of U1. First select **WIRE** and draw a signal wire from U1, pin 10 as shown in figure 1-5. Next, select **OBJECT** and enter:

GND <Enter>

A ground symbol appears. Position the ground symbol in the Drawing Window and place it at the end of the U1, pin 10 wire, as shown in figure 1-5. Ground is now assigned to U1, pin 10.

Next, draw a signal **WIRE** from U2, pin 4 as shown in figure 1-5. We will now connect a resistor to the end of this wire. Select **DRAW-OBJECT** and enter:

DRES\2,R1,1K,10%,1W,050-1000-025 <Enter>

Place the resistor. Be sure to align the resistor stub with the exact end point of the wire, as shown in figure 1-5.

Notice that the **DRAW-OBJECT** Parameter String entry for the resistor included four component-specific parameters after the reference designator: **1K,10%,1W,050-1000-025**. The fourth parameter, **050-1000-025**, is the resistor's part number. This parameter (part number) is never

displayed with the component, but is used in the drawing's SCHPOST bill of materials output. Part numbers for SCHEMA III digital components are assigned in the Object Editor, described later. Most of SCHEMA III discrete component objects are assigned parameters through the **DRAW-OBJECT** Parameter String. For example, capacitors normally carry value and voltage rating parameters, while a crystal may require a frequency parameter and so on. The actual physical location of these parameters (with respect to the associated object) is determined in the Object Editor when the component is created. The Schematic Editor **EDIT-PM OBJ** command identifies the format of a given component's **DRAW-OBJECT** Parameter String. Select **EDIT-PM OBJ** and position the cursor over resistor R1. Now either press **<Ins>** or the mouse button. The Text Window appears with the resistor's **DRAW-OBJECT** Parameter String format:

,REFD,VALUE,%%%%,WATT,PART_NO

This is the **DRAW-OBJECT** Parameter String entry format for resistors. Note that the **PM OBJ** command simply identifies the string *format* and does not allow for editing the parameters or their position within the string. This can only be done using the Object Editor. Any time a component is used, you can use the **EDIT-PM OBJ** to identify the proper **DRAW-OBJECT** Parameter String entry format.

Now draw a signal wire from the left side of resistor R1 to the left about 3 visual grid units or so. We will now assign a label of VCC to the input of resistor R1. Select **DRAW-LABEL**. Select the **LABEL** Command Box until the icon reflects a label to the left of the wire. Now, enter:

VCC <Enter>

and place the VCC label at the left-most end point of the wire we just connected to R1. Check your work against figure 1-5.
We will now connect U2, pin 1 to U2, pin 4. Draw a signal wire from U2,

pin 1 as shown in figure 1-5. Next, complete the connection by extending the wire up to U2, pin 4. Now select **DRAW-DOT** and position the cursor in the Drawing Window. Press **<Ins>** or press and hold your mouse button. A connection dot appears. Position the dot directly on top of the U2, pin 4 signal wire T-intersection, as shown in figure 1-5 and repress **<Ins>** or release your mouse button to place the dot.

DOTs inform SCHPOST's post-processors of signal wire T-connections or 4-way connections. Only *signal wire connection junctions with at least three wires* are recognized as legitimate wire connections and will not cause a Routing error in SCHPOST. The wire must be broken at the junction with the **DOT**.

Paragraph 3.3.4.3 of Chapter 3 describes all the legitimate connections recognized by the SCHPOST post-processor programs and the techniques used to make these connections.

Now route a signal wire from U2, pin 5, right about 5 visual grid units. Select **DRAW-LABEL** Command Box until the icon reflects a label to the right of the wire. Now enter:

OUT <Enter>

Then place the OUT label at the right end of the wire as shown in figure 1-5.

Use Object Editor to Create New Component

We will now take a break from drawing in the Schematic Editor and use the Object Editor to make a new component. The new component will then be added to the LOGIC schematic. Before we move on to the next lesson lets save LOGIC. Select **FILE-SAVE**, the text window should display LOGIC.P01. Then press **<Enter>**.

<Left Blank>

LESSON 4- CREATE A PART USING THE OBJECT EDITOR

1.5.4.4 The Object Editor

To invoke the Object Editor, select **FILE** followed by **ED OBJ**. Follow the prompt by pressing **<Enter>**.
The Schematic Editor is replaced with the Object Editor shown in 4-1. The Object Editor is used to create new objects and edit existing objects for use in the Schematic Editor. Chapter 3, *SCHDRAW*, contains additional information on objects.

Object Editor commands are similar to Schematic Editor commands, except they are used exclusively on objects.

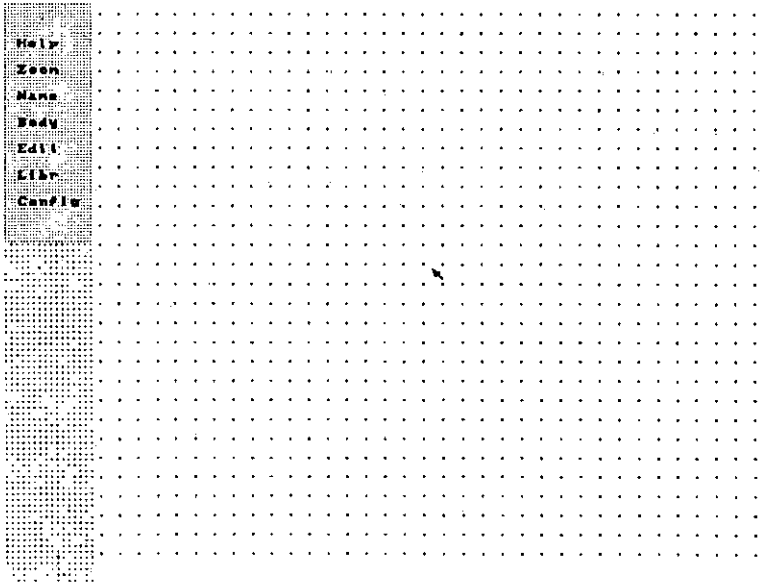


Figure 1-6. The Object Editor

Creating and editing schematic component objects is extremely simple. SCHEMA has three distinct categories of Library Parts: *NESTED* Object, *BODY* Object, and *NAME* Object. The *NESTED* Object is a building block of the *BODY* Object. For example, the *NESTED* Object, OAND is an AND gate symbol with no pin definitons. It's used by *BODY* Objects: T08-1A, T08-1B, T08-1C, and T08-1D.

Many different *NAME* Objects share the same schematic symbol. For example: 74LS08, 74S08, and 74F08 share the same *BODY* objects. The *NAME* Objects have up to four Description Fields and also contain a *BODY* Field, which is a maximum five character prefix taken from the *BODY* Object to which it points. For example, the *BODY* Fields for 74LS08 is T08. The Object Editor commands that pertain to Name Objects are accessed via the **NAME** command, while Body Object commands are accessed through the **BODY** command.

1.5.4.4.1 What is a Name Object?

Name Objects may at first be confusing because in the Object Editor, they appear as lines of text with no visual information or shapes, yet in the Schematic Editor, Name Objects appear as real components with visual shapes, reference designators, pinouts, etc. Admittedly a little strange, but once you understand the power behind Name Objects, you'll further appreciate SCHEMA III and its sophisticated object hierarchy. Let's look at a typical digital Name Object in the Object Editor (analog Name Objects are discussed later). Select **LIBR** followed by **LOAD**. The Text Window appears. Enter:

74LS160 <Enter>

After a second or two, the 74LS160 Name Object is loaded into the Drawing Window (you may have to pan the window slightly to see the object). Notice that the Name Object, shown in figure 1-7, contains just three lines of information.

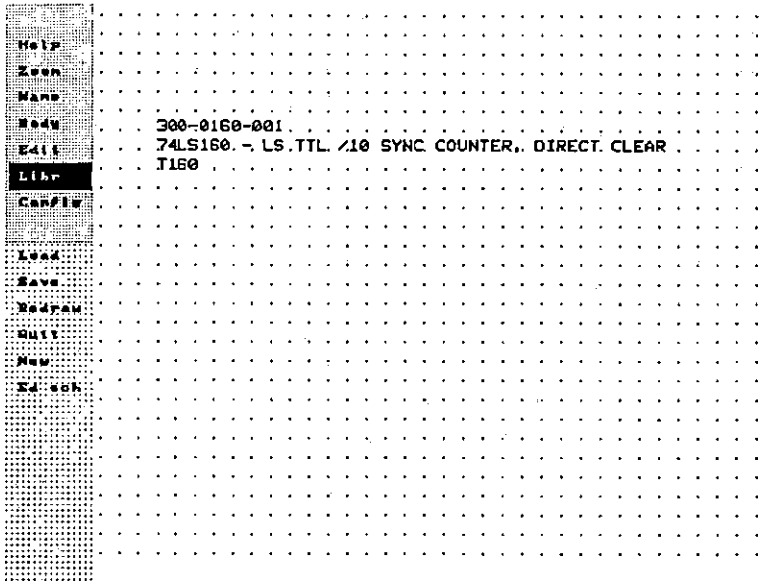


Figure 1-7. 74LS160 Name Object

The first line:

300-0160-001

represents the *Omation* part number for a 74LS160 part. This part number was created by Omation and is probably not the same part number for a 74LS160 that your company uses. Appendix F contains an explanation of the Omation part numbering system used with SCHEMA III components. Most SCHEMA III users will want to change this field to match their own company's part numbering system. To do this, select the NAME-DESC command. The DESC Command Box lights and

displays DESC 1. The Text Window appears with the 300-0160-001 value. Press the <Esc> key to clear the Text Window. Now enter the part number your company uses for a 74LS160 component. When you are finished, press <Enter>. Next, position the cursor in the Drawing Window and press <Ins> or press and hold the mouse button. The new company part number appears in the Drawing Window and as long as you keep the mouse button pressed, you can move the number anywhere in the Drawing Window.

Position the part number above the other two information lines (actually, placement is not critical) and release the mouse button or repress <Ins>. You have just assigned your company part number to the 74LS160 component. This part number will now appear on your SCHPOST bill of materials and net/pin list outputs whenever a 74LS160 component is used in one of your schematics.

The next line of information in the 74LS160 Name Object is:

74LS160 - LS TTL /10 SYNC COUNTER, DIRECT CLEAR

This line simply identifies the component and contains a brief description of it. You may not want to edit this field if you feel this is an adequate description of a 74LS160 component. On the other hand, if you want to change this description, reselect the DESC Command Box. The DESC Command Box (already lighted) increments to DESC 2. The Text Window appears with the 74LS160 component description. Press <Esc> to clear the Text Window and enter desired description of a 74LS160. Upper or lower case is available when using the DESC command. Next, press <Enter>. Now press <Ins> or press and hold the mouse button and place the part description below the company part number (DESC 1) field (again, physical placement within the Drawing Window is not critical). This 74LS160 description field appears on your SCHPOST bill of materials and net/pin lists everytime a 74LS160

component is used in your SCHEMA III schematics.

The two Name Object fields we've examined, **DESC 1** and **DESC 2**, are only used for the SCHPOST bill of materials and net/pin list post-processors. These two information lines (the part number and description) can be ignored if you do not plan to print bill of materials from your schematics. However, most users want SCHPOST to create bills of materials for their schematics. Thus, one of the first things you need to do (assuming you want bill of materials for your schematics) is to change the **DESC 1** part numbers assigned by Omation to reflect the actual part numbers your company uses for every component that you plan to use on your schematics.

As we have just seen, editing the **DESC 1** field is very simple in the Object Editor. But there is an even easier (faster) way to edit/create these information fields. The Library Manager program, SCHLIB, provides a very quick way to edit/create Name Objects and is most useful when editing or creating a large number of Name Objects. Please refer to Chapter 6, *SCHLIB*, and paragraph 6.4.3 in particular.

We have not mentioned the **NAME-DESC 3** and **NAME-DESC 4** fields which are not used with the 74LS160 Name Object and in fact, are shipped with SCHEMA III undefined. These two fields can be assigned to contain any component-related information you want included on your bills of materials/net/pin lists. For example, the **DESC 3** field could contain component cost or pricing data, while the **DESC 4** field could contain manufacturing or vendor-related information, or vice versa. Alternately, these fields could contain per-part current consumption data or even X/Y coordinate data required by simulation/PCB layout/wire wrap programs driven by the SCHPOST net/pin list output.

The last (bottom) information line of the 74LS160 Name Object is the most important piece of information in the Name Object. It is extremely

important that you understand the concept it represents. This line:

T160

is the **NAME-BODY** field, which references the Body Object used by the 74LS160. The Body Object (described in upcoming paragraphs) is the actual, visual shape that represents the 74LS160 component in the Schematic Editor. You will probably not *edit* many existing **BODY** fields. You will, however, *create* many **BODY** fields (and thus, Body Objects) for the new components you create. When entering **BODY** field values, always enter just the **BODY** field prefix (such as T160, T00, T07, etc.). Do *not* include the applicable Body Object's orientation/gate section suffix (T160-1A, T00-1B, T07-1C, etc.) after the **BODY** field.

Name Objects are always saved (**LIBR-SAVE**) as the actual component name, i.e., 74LS160, 80186, LM555, etc. When you add components to your drawings in the Schematic Editor, you use the **DRAW-OBJECT** command and enter the applicable Name Object nomenclature (such as 74LS160, 80186, LM555, etc.) as the first entry of the **DRAW-OBJECT** Parameter String. Let's say you are in the Schematic Editor and you select **DRAW-OBJECT** and enter **74LS160**. SCHDRAW loads the 74LS160 Name Object, finds the **BODY** field, **T160**, and automatically loads the default orientation (-1A) T160 Body Object into the Schematic Editor. This explains why Name Objects in the Object Editor appear as lines of text but are displayed as actual, visual components in the Schematic Editor.

1.5.4.4.2 Body Objects

Body Objects are the visual shapes that represent components. Body Objects can only be created and edited in the Object Editor using the **BODY** Secondary Commands (unlike Name Objects which can be created/edited from the Object Editor and/or the SCHLIB program). The only way to really understand Body Objects is to load one and

examine its characteristics. To do this, select **LIBR-LOAD.** The Text Window appears. Now, enter **T160-1A,** which is the standard, default orientation of a 74LS160 component, followed by **<Enter>.**

NOTE

When you load Body Objects, you must always include the object's complete prefix and orientation/gate section suffix (such as T00-1B, T160-1A, T07-2B, etc.).

After a second or two, the 74LS160 Name Object disappears and the T160-1A Body Object appears in the Drawing Window (you may have to pan the Drawing Window slightly to see the object).

The T160-1A Body Object (figure 1-8) consists of the following:

An IC shape
Pin Headers (**BODY-HDR**)
Pin Names (**BODY-TEXT**) such as LD, CL, QD, etc. and Pin Numbers such as 1, 2, 3, etc.
The Reference Designator Parameter (**BODY-PARM R**): REFD
The Name Parameter (**BODY-PARM N**): NAME...
Pin Definitions (**BODY-DF PI**) such as I1, I2, O3, etc. and Power and Ground Assignment Lines such as I16 | @+5 and I8 | @GND.

The IC shape was drawn using the **BODY-LINE** command. The LINE command provides several different line types: thin, fat, dotted, or dashed. Curved lines, your company logo, and other unique symbols/shapes are created using the BIT command, discussed later.

Pin Headers (stubs) were added to the square IC shape using the **BODY-HDR** command. Several pin header types and orientations are available by repeatedly selecting the **HDR** Command Box.

Pin Names (**CL, LD, QD,** etc.) and Pin Numbers (**1, 2, 3,** etc.) were added using the **BODY-TEXT** command. All TEXT information is ignored by SCHEMA III post-processing programs but is displayed with the object in the Schematic Editor *in the exact form and position it is entered in the Object Editor.* That's really all there is to pin names and numbers. (Vertical TEXT is created by entering a backslash character (\), followed by a **<Space>** and the applicable TEXT.). Some users may find that pin name/number placement is made easier by turning off the Cursor Grid (**CONFIG-C GRID** Command Box extinguished). With **C GRID** off, you have pixel-by-pixel cursor resolution and can place **TEXT** anywhere in the Drawing Window. (When **C GRID** is on, the mouse cursor is always snapped to an invisible 8-pixel grid, which restricts cursor movement to on or half-way between the visual grid dots.).

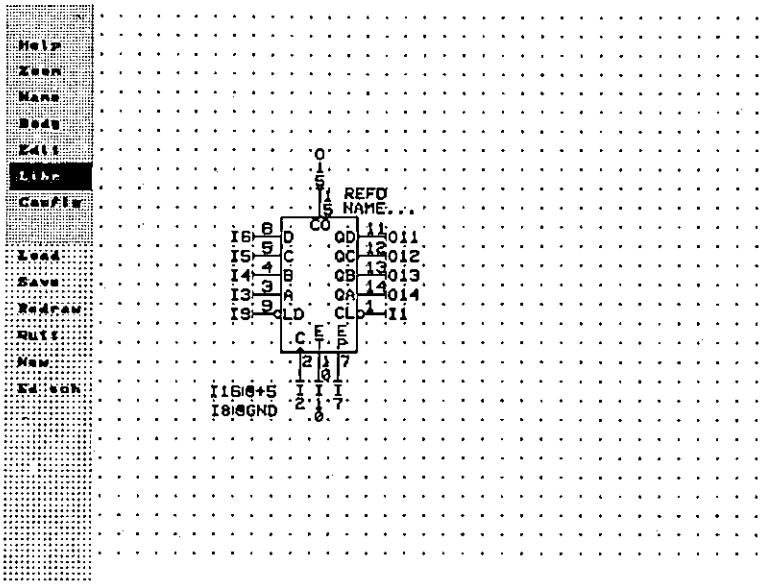


Figure 1-8. T160-1A Body Object

The **BODY-PARM R** Reference Designator Parameter, **REFD**, informs SCHEMA III to literally replace **REFD** with the component's reference designator (the REFD is auto-assigned by the Schematic Editor or entered as part of the **DRAW-OBJECT** Parameter String) when the component is used in the Schematic Editor. Thus, when the 74LS160 component is used in the Schematic Editor, the **REFD** field is replaced with the component's actual reference designator.

Figure 1-9 illustrates the T160-1A Body Object, as shown in the Object Editor, versus the 74LS160 component shown in the Schematic Editor. Notice that **REFD** has been replaced with reference designator **U1**.

NAME... is the Name Parameter (**BODY-PARM N**) which is similar to the **REFD** field except **NAME...** is replaced with the actual name of the associated Name Object when the component is used in the Schematic Editor. For example, **NAME...** is literally replaced with **74LS160** when this object appears in the Schematic Editor. See figure 1-9. Since Name Objects are saved as the actual component name, **NAME...** is replaced with the actual component name, i.e., 74LS160, LM555, 8048, etc. in the Schematic Editor. The three periods after NAME (...) act as pad values for the length of the Name Object. For example, if you just created a Body Object for a 74HC4050 component and had previously saved its associated Name Object as 74HC4050 (eight characters long), the Name Parameter, **PARM N**, would have to be **NAME....** *or eight total characters* (NAME plus four periods (....) to equal the length of 74HC4050). Thus, **NAME....** would provide enough room to display 74HC4050 in the Schematic Editor.

There are four additional **BODY-PARM** command fields that we have not discussed: **PARM 1, PARM 2, PARM 3,** and **PARM 4.** These are normally used with discrete components to denote part value (such as 10K, 100uF, etc.), power ratings (such as 1W), tolerance (1%, 5%, 10%, etc.), the company part number, etc. These parameters are discussed in detail in an upcoming paragraph.

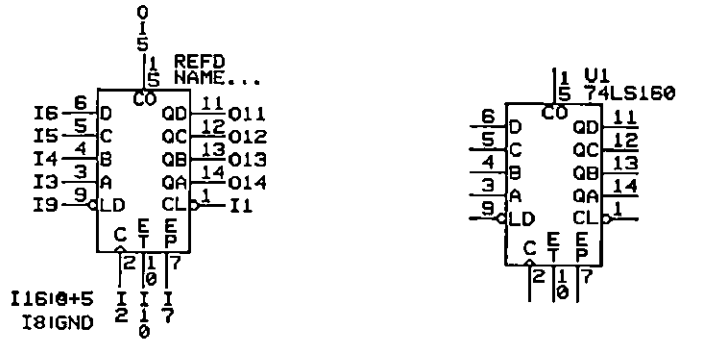


Figure 1-9. T160-1A Object Editor Body Object vs 74LS160 Schematic Editor Component

Pin Definitions (**BODY-DF PI**) **I6, I5, O14,** etc. inform SCHEMA III SCHPOST program net/pin/wire list and DRC post-processors of pin type and number such as input, output, don't care, etc. Pin definitions are not displayed with objects in the Schematic Editor - see figure 1-9. The pin definition format is:

Axxx

where **A** is the pin type and xxx is normally the pin number. Legal **A** values include:

I Input
O Output
T Tri-state Output
B Bidirectional
C Open Collector
@ Absolute
X Don't Care (pin not used for DRC).

Pin definitions should be in uppercase. Lowercase letters will be automatically converted to uppercase when the SCHPOST function is invoked.

Use the Define Pin (DF PI) command to assign pin definitions to Body Objects. There are four possible pin definition orientations: left, right, up, and down. Select the DF PI Command Box several times. You will notice that the DF PI Command Box icon, consisting of a small box and line, toggles back and forth as the command box is selected. This icon represents the relationship between the pin definition you enter (represented by the small box) and the pin header or stub (represented by the line) to which the definition attaches.

The power and ground assignment lines:

I16 | @+5 and **I8 | @GND**

were also assigned using the **DF PI** command. The procedure to assign power and ground to components is described later.

NOTE: There is a new Object Editor Command **BODY-PIN** that automates the pin and hidden pin creation. An example will be supplied in the section **,1.5.4.4.8.1 Tutorial Body Object**

Component Body Objects must always be saved with a complete body prefix and orientation/gate section suffix. You may follow Omation's naming convention or create your own. Appendix F, *Object Libraries*, contains a complete description of Omation's Body Object naming convention. Keep in mind that the length of the complete Body Object name, including suffix, must not exceed eight characters (limited by DOS).

1.5.4.4.3 Discrete Name and Body Objects

Discrete Name and Body Objects differ slightly from their digital counterparts.

Let's load a discrete Name Object. Load the SCHEMA III Name Object for a resistor by selecting **LIBR-LOAD.** The Text Window appears. Enter:

DRES <Enter>

After a few seconds, the DRES (resistor) Name Object appears in the Drawing Window (you may have to pan the Drawing Window slightly to see this object). Figure 1-10 shows the DRES Name Object. Like the 74LS160 Name Object, DRES has three fields of information: the **DESC 1, DESC 2,** and **BODY** fields. The **DESC 1** field value:

<P4>

informs SCHPOST to use the component's **DRAW-OBJECT** Parameter String fourth parameter (**<P4>**) entry and literally print the entry on its bills of materials/net/pin lists.

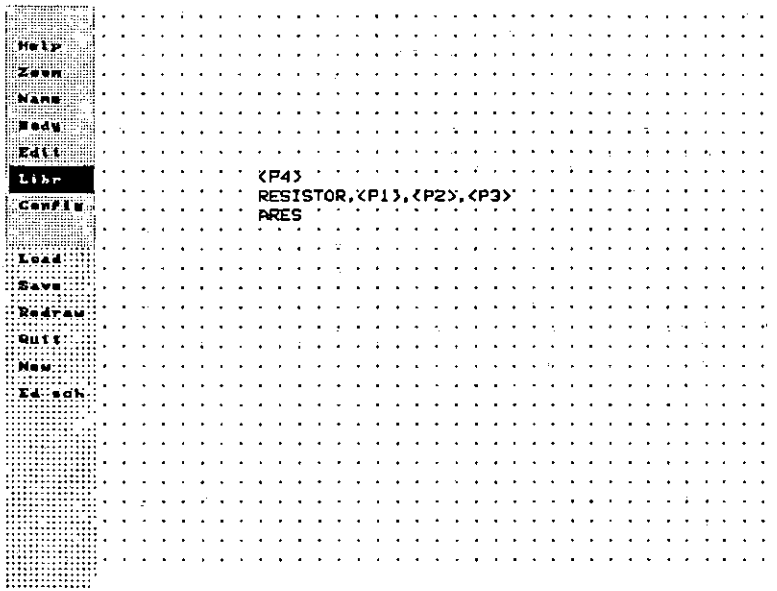


Figure 1-10. DRES Name Object

Thus, if you want your company part number for a resistor to print on your bill of materials, you must include this part number every time you call up a resistor in the Schematic Editor using the **DRAW-OBJECT** command. Confused? Let's briefly review parameters and their use.

Every SCHEMA III component used in the Schematic Editor is retrieved by entering a **DRAW-OBJECT** Parameter String (figure 1-11). The **DRAW-OBJECT** Parameter String can contain up to six parameter (PARM) entries:

NAME,REFD,<P1>,<P2>,<P3>,<P4>

The first entry, NAME, is the Name Object parameter which is entered every time you use the **DRAW-OBJECT** command to retrieve an object. For example, to add a 74LS00 object to the current schematic, you select the Schematic Editor **DRAW-OBJECT** command and enter the Name Object parameter: **74LS00 <Enter>**. You could also include the reference designator (**REFD**) parameter, by entering **74LS00,U1 <Enter>**. Upper or lower case may be used for the **NAME** and **REFD** fields. The letters will be automatically converted to uppercase when they appear on the screen. Upper or lowercase is available for the **P1 - P4** fields. The letters will appear in the same case as entered. This would retrieve a 74LS00 with an assigned reference designator of U1. Note that each parameter field is separated by a comma. As discussed many times in this manual, every time you do not include the reference designator parameter when retrieving a component using **DRAW-OBJECT**, the Schematic Editor automatically assigns one for you (.AAA01, .AAB01, .AAC01, etc.). As demonstrated earlier in the tutorial, auto-assigned reference designators are edited using the Schematic Editor **EDIT-IMAGE** command.

The remaining four parameters (**<P1>, <P2>, <P3>, <P4>**) are user-defined. Thus, any component-related information you want displayed with the component or printed on the bill of materials output can reside in these four fields.

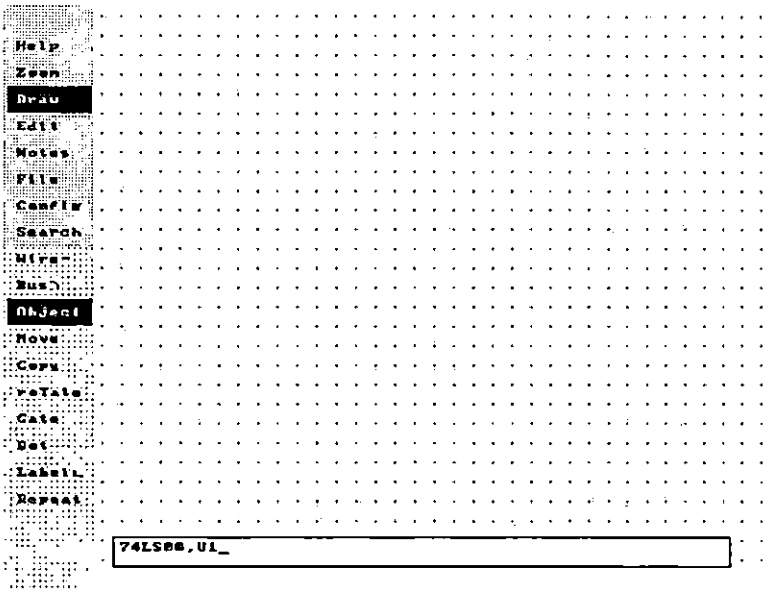


Figure 1-11. 74LS00 DRAW-OBJECT Parameter String

Omation shipped **PARM 4 (<P4>)** predefined as the company part number field for discrete components. This can be left alone or changed to suit your requirements. Omation also predefined the **PARM 1, 2,** and **3** fields for resistors and some other discrete components. It is your responsibility to include these fields (with the proper data) within the component's **DRAW-OBJECT** Parameter String entry *every time you use these components in the Schematic Editor.*

For resistors, **PARM 1 (<P1>)** is the part **VALUE**, in Ohms (1K, 50K, 33K, etc.), **PARM 2 (<P2>)** is the part's tolerance, *%%%%*, as a percentage (1%, 5%, 10%, etc.), and **PARM 3 (<P3>)** is the part's power rating, **WATT**, in watts (1/2W, 1W, 2W, etc.).

Obviously, **DRAW-OBJECT** Parameter String entries/values differ from discrete part to discrete part. For instance, a crystal may have a frequency entry while capacitors may have a voltage rating entry, etc. When you create new components, define these parameters to match individual component requirements.

OK, let's get back to the DRES Name Object. We have discussed the **DESC 1** field, **<P4>**. Now let's look at the **DESC 2** field:

RESISTOR, <P1>,

As with digital Name Objects, the **DESC 2** field is the part description printed on your BOMs. Thus, **RESISTOR** will be printed on your BOMs followed by the **<P1>** parameter field (the resistor **VALUE**, in Ohms), entered as part of the resistor's **DRAW-OBJECT** Parameter String. The **DESC 2** field could be modified to include the resistor's tolerance and wattage rating by simply adding a **<P2>** and **<P3>** reference. For example, let's say you modify the **DESC 2** entry to be:

RESISTOR,<P1>,<P2>,<P3>

and you add a 10K, 1%, 1/2W resistor to your schematic. The corresponding SCHPOST bill of materials output would then show:

RESISTOR, 10K, 1%, 1/2W

for this resistor under the appropriate column on your bill of materials.

The **BODY** field, **ARES**, simply points to the two resistor Body Objects: ARES-1A and ARES-2A. This operation is identical to digital Name Objects.

Now let's load the vertical resistor Body Object, ARES-1A. Select **LIBR-LOAD** and enter:

ARES-1A <Enter>

After a few seconds the DRES Name Object disappears and the vertical resistor Body Object, ARES-1A, appears in the Drawing Window (you may have to pan the Drawing Window slightly to see the object).

As shown in figure 1-12, ARES-1A consists of the following:

A resistor body shape
Two pin stubs
PARM R, - REFD;
PARM 1 - VALUE; **PARM 2** - %%%%;
PARM 3 - WATT
Pin definitions (**DF PI**) - XT and XB.

The resistor body shape was drawn dot-by-dot (pixel-by-pixel) using the **BODY-BIT** command, discussed later.

The two pin stubs were added using the **BODY-LINE** command. Alternately, vertical **HDRs** could have been used.

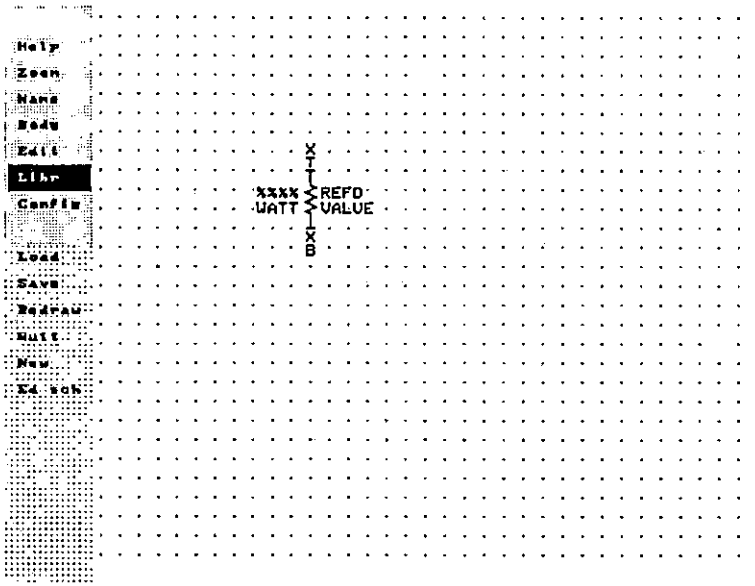


Figure 1-12. ARES-1A Body Object

The Reference Designator Parameter (**PARM R**), **REFD**, is used as described in previous paragraphs. **PARMs 1, 2, and 3** (**VALUE**, *%%%%*, and **WATT**), are used in a similar manner. **VALUE** is replaced with the resistor's value (in Ohms) entered as part of the **DRAW-OBJECT** Parameter String in the Schematic Editor. *%%%%* is replaced with the resistor's tolerance (as a percentage) also entered as part of the Schematic Editor **DRAW-OBJECT** Parameter String entry. **WATT** is replaced with the resistor's wattage rating, again, entered as part of the **DRAW-OBJECT** Parameter String. (The Schematic Editor **EDIT-PM OBJ** command is used to identify these parameters from the Schematic Editor. Note that **EDIT-PM OBJ** only *identifies* these parameters; you can only edit them from the Object Editor.).

Pin definitions **XT** and **XB** were assigned with the **BODY-DF PI** command. As previously discussed, **X** is the Don't Care prefix and will be ignored by SCHEMA III DRC post-processor. The **T** and **B** suffixes equate to Top and Bottom to physically define the top and bottom resistor connections for the SCHPOST net/pin/wire list post-processors. (Horizontal resistor orientations use **XL** and **XR**, for Don't Care/Left and Right). Obviously, there are no power and ground assignment lines for a resistor.

1.5.4.4.4 Nested Objects

A most powerful feature of SCHEMA III is its ability to *nest* objects. Many separate Nested Objects can actually make up a single Body Object.

Any time you want to add (nest) a Nested Object to a Body Object, use the **BODY-N OBJ** command. **N OBJ** is used exclusively to add Nested Objects to Body Objects. Never, repeat, *NEVER* use the **LIBR-LOAD** command to nest objects. **LIBR-LOAD** simply erases the Drawing Window contents and loads an object. Only use **N OBJ** to add Nested Objects to other objects.

When **N OBJ** is selected, the Text Window appears to let you enter the name of the Nested Object you want to add to the current object. Enter the applicable Nested Object name followed by **<Enter>**. Next, position the cursor in the Drawing Window and press **<Ins>** or press and hold the mouse button. The Nested Object appears and can be moved anywhere in the Drawing Window as long as the mouse button is held pressed. To place the object, repress **<Ins>** or release the mouse button.

The **N OBJ** command adds Nested Objects as whole entities. Thus, you will not be able to modify preexisting **PARMs, LINEs, HDRs,** etc. on Nested Objects retrieved using **N OBJ** (use **LIBR-LOAD** to edit these). For example, if you try to **EDIT-ERASE** a given part of a Nested Object that was retrieved using **N OBJ** the entire object will be erased. Of course, you can *add* **PARMs, DF PI** values, **HDRs, LINEs, TEXT, BITs,** etc. to Nested Objects.

Nested Objects can generally contain the same subcomponents as Body Objects (**LINEs, TEXT, HDRs, BITs,** etc.). Nested Objects can not, however, contain **DF PI** entries (pin definitions and power/ground assignment lines). The exact combination of subcomponents used to compose a given Nested Object is determined by you and to some extent, by the object itself.

A good example of a Nested Object is the object called ONAND. ONAND is a NAND-gate shape that has been nested by Omation within many Body Objects. One such Body Object is T00-1A. This Body Object is used over and over again to represent the A-gate section of many standard IC components such as 74ALS00, 74F00, 74HC00, 74LS00, etc. **N OBJ** is an *extremely powerful* command. Experiment with **N OBJ** until you thoroughly understand its operation.

1.5.4.4.5 The BODY-BIT Command

The **BODY-BIT** command lets you create/edit images pixel-by-pixel. These images can then be added to any Body/Nested Object or saved as Body/Nested Objects themselves. The incredible editing resolution provided by **BIT** simplifies small, detailed, unique object creation and editing. The OMATION logo (visible in the title block of SCHDRAW's blank, bordered pages) was created entirely using the **BIT** command. Let's look at the OMATION logo to get acquainted with the **BIT** command.

First, turn off the cursor grid (C GRID) by selecting the CONFIG-C GRID Command Box or pressing **<F10>**. The C GRID Command Box extinguishes. Next, load the OMATION logo Body Object by selecting **LIBR-LOAD** and entering:

OMATION1 <Enter>

(The OMATION1 Body Object does not have or require a Name Object since it does not carry any electrical, schematic-oriented information). After a second or two, the OMATION1 Body Object loads (you may have to pan the Drawing Window slightly to see the object). Next, select the **BODY-BIT** Command Box. The **BIT** Command Box lights and displays **BIT 1.**

Now position the cursor directly on top of the OMATION1 Body Object and click the mouse button or press **<Ins>**. The **BIT** Command Box increments to **BIT 2** and the Drawing Window displays a blown-up portion of the OMATION1 Body Object in the *Bit Window.*

The Bit Window displays a 68x40 pixel bit map. Each pixel is displayed as a visual dot. The portion of the OMATION1 Body Object shown in the Bit Window can be edited by positioning the cursor over the desired pixel and pressing **<Ins>** twice or clicking the mouse button to add the pixel. Selecting an existing pixel will erase it. To return to the Drawing Window and place the bit image, reselect the **BIT** Command Box. The **BIT** Command Box increments to **BIT 3** and the normal Drawing Window appears. Position the cursor in the Drawing Window and press **<Ins>** or press and hold the mouse button. The OMATION1 bit image appears. As long as the mouse button is held pressed down, the bit image can be moved throughout the Drawing Window. To place the bit image, repress **<Ins>** or release the mouse button. To add a copy of the same bit image, repress **<Ins>** or the mouse button and place as previously described.

To create a new bit image, select the **BIT** Command Box until **BIT 2** is displayed. The Bit Window appears. Position the cursor and create the bit image by pressing **<Ins>** or clicking the mouse, as required. When finished, select the **BIT** Command Box to **BIT 3** and place the image in the Drawing Window, as previously described.

NOTE

Use BIT to either create new bit images or edit predrawn bit images. You cannot use BIT to edit non-bit images such as LINEs, HDRs, TEXT, etc.

Obviously, the possibilities the **BIT** command provides are endless. Omation used the same SCHDRAW Object Editor you have to create all the Body Objects that were included in your SCHEMA III package. Omation used the **BIT** command any time unique shapes were required such as arcs, dots, circles, bubbles, gate symbol curves, arrows, etc. Use **BIT** to draw discrete components, your company's logo, or any unique shape/object. The **BODY-LINE** command can also be used to draw circles, arcs, and diagonal lines - refer to the SCHDRAW command summary in paragraph 3.5 of Chapter 3 for additional information.

1.5.4.4.6 The EDIT-ID Command

The **EDIT-ID** command is used to identify object subcomponents. Remaining **EDIT** Secondary Commands operate identically to the Schematic Editor **EDIT** commands, discussed later. To use **EDIT-ID**, select **EDIT** followed by **ID**. The **ID** Command Box lights. Next, position the cursor over a portion of the desired object/image and press **<Ins>** or the mouse button. The Text Window appears with a precise identification of the associated object/image part.

ID is very handy when you are editing objects and you need to know the make-up of a particular object. You will be able to quickly identify bit

images, nested objects, LINE/TEXT, etc. within objects. Table 3-2 in Chapter 3 contains a complete listing of ID displays and their meanings. Use ID to explore and understand the techniques Omation used to create objects - it will greatly aid/enhance your ability to create/edit your own objects.

ID use may be facilitated by turning off the C GRID. Also, it may be necessary to press <Ins> or your mouse button several times to ID all object subcomponents at a given cursor position.

1.5.4.4.7 Remaining Object Editor Commands

Remaining Object Editor commands include EDIT, CONFIG, HELP, LIBR, and ZOOM. These commands operate identically to their equivalent Schematic Editor commands. The EDIT-UPDATE command operates like the Schematic Editor EDIT-IMAGE command except it is used to edit BODY-TEXT and BODY-DF PI entries. The LIBR commands operate identically to the Schematic Editor FILE commands except LIBR-NEW which simply clears the Object Editor Drawing Window and LIBR-ED SCH which invokes the Schematic Editor.

1.5.4.4.8 Creating the Tutorial Component

We will now create a simple, digital IC component to add to the LOGIC schematic. We will create the Body Object first, although we could just as easily create the Name Object first.

1.5.4.4.8.1 Tutorial Body Object

First, ensure **CONFIG-V GRID, C GRID** and **WI HDR** are lighted. Also ensure the Object Editor Drawing Window is empty by selecting the **LIBR-NEW** command. After a few seconds, the Drawing Window is initialized (**NEW** Command Box extinguishes). The completed Body Object is shown in figure 1-13. The following paragraphs provide a step-by-step procedure to draw the tutorial Body Object.

Previous versions of SCHEMA required the user to create pins in a piece by piece fashion. Using **BODY-TEXT** to describe the pin. Using **BODY-HDR** to select a standard pin or a bubble output. Then using **BODY-TEXT** to define the pin number and the internal text label of the pin. Lastly, use **BODY- DF PI** to define the pin definition. However, now there is a new command that automates this process and yields a cleaner, more consistent pin definiton. This command is the **BODY - PIN** command. The old commands and methods still function the same but the new command automates it by creating a grouping of the text and definitons together for placement. Once placed all the sub-components are then separate. You can use **EDIT-ID** later to see from where these sub-components are derived.

Add Pins and Draw the Component Shape

Select the command **BODY-PIN.** Click this command light several times and you will see all the different icons. The square box or the bubble of the icons represents the orientation of the pins and their definition in relation to the body of the part. The '@' icon indicates a hidden pin. See the following figure:

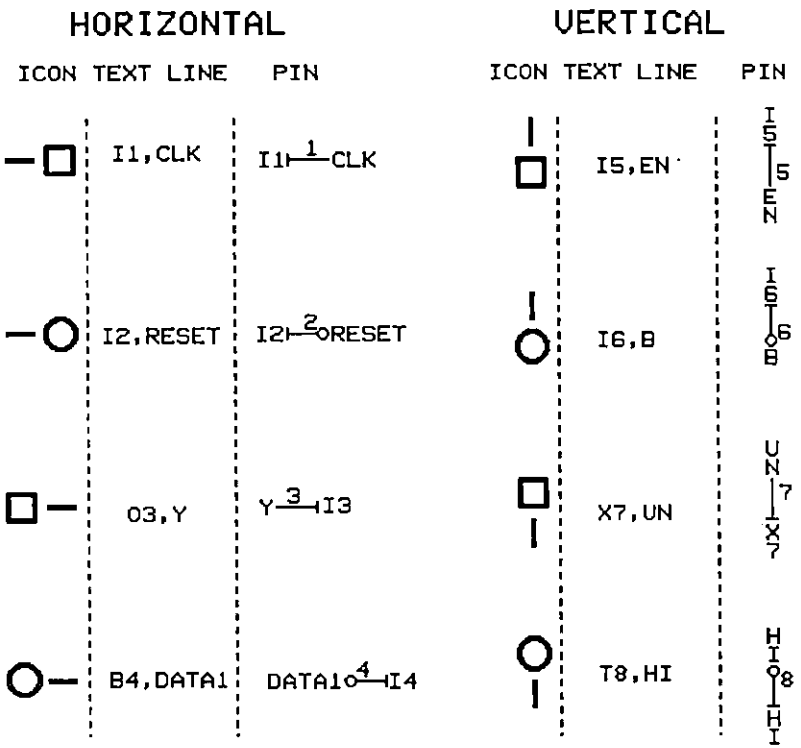


Figure 1-13A BODY-PIN Icons

Select the **BODY-PIN** icon with a box to the right of the line.

In the Text Window type: **I1,IN <Enter>**

Then move the cursor to the center of the screen and place the pin down. You will notice that the pin header has the pin definition ,"I1", which means that it's an input and its number is 1. The pin number is also placed above the header. This is necessary since the actual pin definition will not be shown with the part in the Schematic Editor. Lastly is the **BODY-TEXT, "IN"**, that is a optional extra information for the user. Next, select the **BODY-PIN** icon with the box on the left. Now we will create the output pin.

In the Text Window type: **O2,OUT <Enter>**

Place this pin to the right of the first pin, where the pin definiton junction bars ' | ' are approximately 8 grid units apart.

To draw the object shape, which will be a box, select the **BODY-LINE** Command Box. The **LINE** Command Box lights with the thin **LINE** icon. Next, position the cursor in the Drawing Window and use the mouse to draw the box. As you draw the box shape, observe the Cursor Coordinate Window in the lower left corner of the Drawing Window to ensure the box is 2 grid units high and that the line passes between the header line and the text, "IN" and "OUT". We have just defined the basic IC component shape.

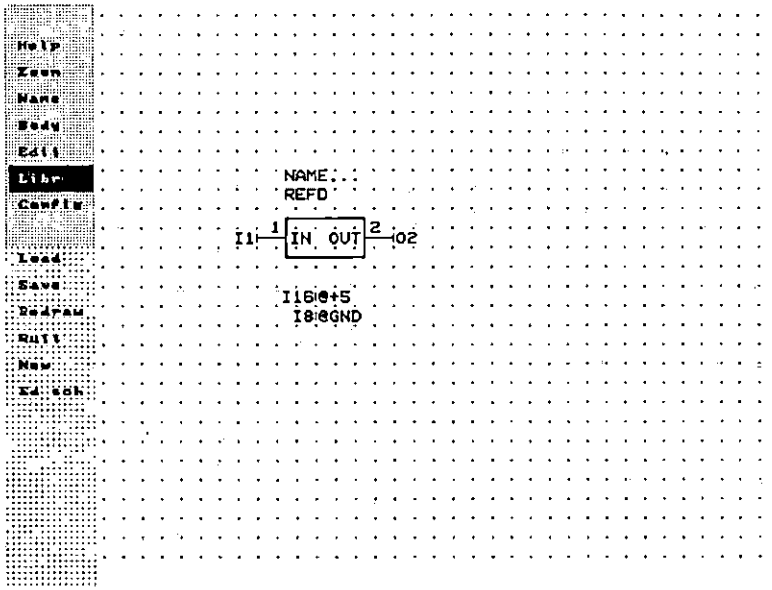


Figure 1-13. Completed Tutorial TXXX-1A Body Object

Define the Hidden Pins for Power and Ground

These hidden pins are seen by the post-processor, **SCHPOST**, however they will be hidden from the user when the part is displayed in the Schematic Editor. Select the **BODY-PIN** command until the icon, '@' is displayed. The '@' is a pin attribute that tells the post-processor that the following string of characters is a net name (Label) not a pin number. In the Text Window type:

I16,+5 <Enter>

Place the hidden power pin under the part. Now type in the Text Window:

I8,GND <Enter>

Place the hidden ground pin under the part as well.

NOTE: When you are in the Schematic Editor and you place a label on a wire with the value, **+5**, this net is incorporated with the hidden net, **+5**. The same is true for the label , **GND**, or using the graphical symbol **,GND**.

Define Parameters and Their Locations

We will now define the reference designator and name parameter replacement fields. Select the **BODY-PARM** command. The Text Window appears and the **PARM** Command Box lights with an N icon, representing the name parameter. Enter:

NAME...<Enter>

Position the cursor in the Drawing Window and either press **<Ins>** or press and hold the mouse button. Now place **NAME...** in the exact position you want to see the object's name displayed when the object is used in the Schematic Editor. Figure 1-13 shows a suitable position. Reselect the **PARM** Command Box. The **PARM** icon advances to **R**, the reference designator parameter.

Now enter:

REFD <Enter>

Place **REFD** in a similar manner as **NAME...** - see figure 1-13. **REFD** is

replaced with the component's reference designator when the component is displayed in the Schematic Editor. Note that the NAME... and/or REFD fields can be placed anywhere near the Body Object, even inside the Body Object, if desired.

Your object should resemble figure 1-13.

Save the Body Object

The only thing left for us to do is save the Body Object. Before we save the object, let's review the procedure we followed to create the Body Object. We first defined the object's pins, using the **BODY-PIN** command. Next we created the object shape using the **BODY-LINE** command. (If we had wanted unusual geometric shapes, we would have used the **BIT** or **LINE** commands.) Then we assigned hiddden power and ground pins to the object. Lastly, we defined the component's parameters and their locations.

Now, let's save the Body Object. Select the **LIBR-SAVE** command. The SAVE Command Box lights and the Text Window appears. Enter:

TXXX-1A <Enter>

where the **TXXX** prefix (**XXX** is normally a numeric field) is the object's body field and the **-1A** suffix defines the object's first orientation and gate section.

Always include an orientation/gate section suffix when saving your Body Objects, *even if there are no other orientation/gate section Body Objects associated with the object.* Use -1A as your default. If we wanted to make a **'B'** gate from this same part, we would use the **EDIT-UPDATE** command to change the I1 to I3, O2 to O4, and the text above the pins respectively. Then save the part as TXXX-1B.

Tutorial Name Object

We must now create a Name Object for the TXXX Body Object. The Name Object consists of several lines of information that carry TXXX-specific part description data to SCHPOST's post-processing programs and associate the TXXX-1A Body Object with the Name Object. To create the Name Object, select the **LIBR-NEW** command (make sure you have saved the TXXX-1A Body Object first). The **NEW** Command Box lights while the Drawing Window is cleared. Figure 1-14 shows the completed tutorial Name Object. The following paragraphs provide a step-by-step procedure to create the tutorial Name Object.

Assign a Part Number to the Component for Post-Processing

Select the **NAME-DESC** command. The **DESC** Command Box lights with **DESC 1** and the Text Window appears. Enter the following sample company part number for the tutorial component:

<Enter>

Place the **050-0010-001** text anywhere in the Drawing Window.

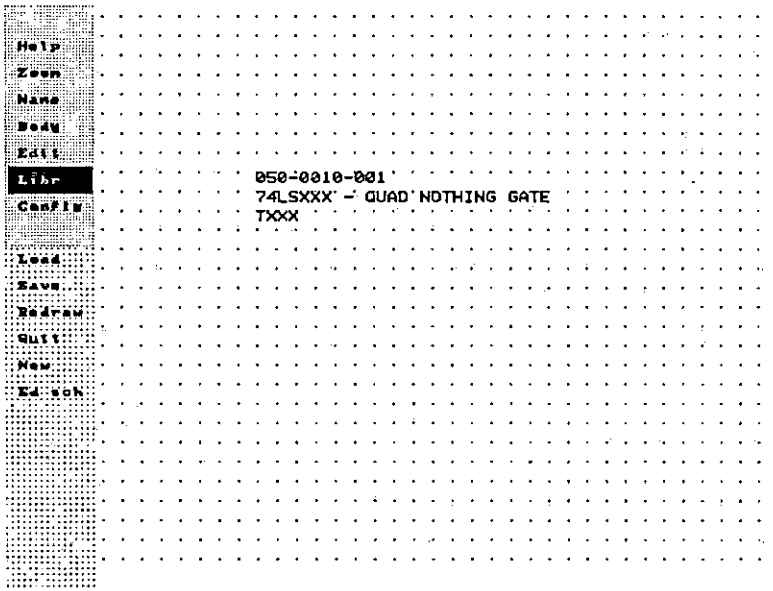


Figure 1-14. Tutorial 74LSXXX Name Object

Enter a Brief Description of the Component for Post-Processing

Re-select the **DESC** Command Box. The **DESC** Command Box remains lighted and increments to **DESC 2**. Enter the following description of the component in the Text Window:

74LSXXX - QUAD NOTHING GATE <Enter>

Place this description below the part number **DESC 1** field, as shown in figure 1-14. These two Name Object fields, **DESC 1** and **DESC 2**, are printed on the SCHPOST bill of materials output when the associated component is used in the Schematic Editor.

Link the TXXX Body Object to this Name Object

Select the **NAME-BODY** command. The **BODY** Command Box lights and the Text Window appears. Enter the associated Body Object prefix name:

TXXX <Enter>

(Do not include the orientation/gate section suffix.). Now place **TXXX** below the **DESC 2** field as shown in figure 1-14.

Save the Name Object

We have completed the Name Object and need to save this Name Object as the actual, generic part name (such as 74LS00, 4164, 8048, etc.). Select the **LIBR-SAVE** command and enter:

74LSXXX <Enter>

The completed 74LSXXX Name Object is saved.

NOTE: If you are creating an object that has only one orientation then you can combine the **NAME-DESC** , parameters with the **BODY** image and save the part to one unified name.

We will now add the 74LSXXX component to the LOGIC schematic in the Schematic Editor. First, return to the Schematic Editor by selecting **LIBR** followed by **ED SCH**. Follow the prompt by entering **<Enter>**. The Schematic Editor loads and displays the upper left corner of the LOGIC drawing. Pan the LOGIC drawing, if necessary, until the LOGIC circuit is visible.

<Left Blank>

LESSON 5- FINISH THE LOGIC SCHEMATIC

1.5.4.5 Completing the LOGIC Schematic

The following paragraphs provide a step-by-step conclusion to the LOGIC schematic tutorial. Be sure the LOGIC.P01 file is loaded.

Add 74LSXXX Component to LOGIC Schematic

Select the **DRAW-OBJECT** command. The **DRAW** and **OBJECT** Command Boxes light and the Text Window appears. Next, enter the name of the component we just created in the Object Editor:

74LSXXX,U3 <Enter>

Position the cursor in the Drawing Window and either press **<Ins>** or press and hold the mouse button. The 74LSXXX component appears in the Drawing Window. Now position the 74LSXXX component below the 74LS00 (U1), as shown in figure 1-15.

Study figures 1-13, 1-14, and 1-15 to compare the Schematic Editor 74LSXXX component with the Object Editor 74LSXXX Name Object and TXXX-1A Body Object. When 74LSXXX was entered under **DRAW-OBJECT**, the Schematic Editor loaded the 74LSXXX Name Object and located the **BODY** field (in the 74LSXXX Name Object). The **BODY** field informed SCHDRAW to load the default Body Object, TXXX-1A.

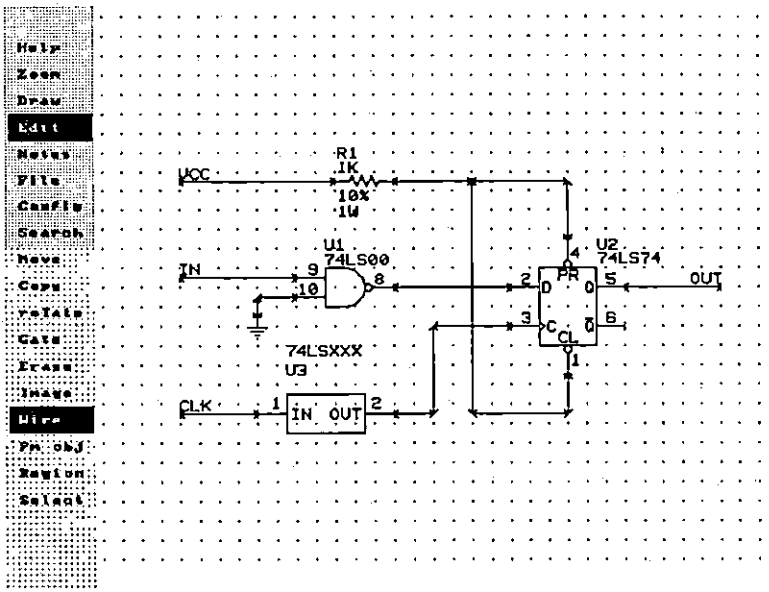


Figure 1-15. Completed Tutorial LOGIC Schematic

The Body Object -1A orientation is always default-loaded into the Schematic Editor if an orientation/gate section suffix is not included in the DRAW-OBJECT Parameter String. For example, if we create a new sub-gate version of the 74LSXXX object, we would create a new Body Object and save it as TXXX-1B to represent this different gate section. Similarly, if we created a new physical orientation for our 74LSXXX, we would save the new Body Object as TXXX-2A, and so on. From the Schematic Editor, under **DRAW-OBJECT**, we could enter **74LSXXX\B** or **74LSXXX\2** to tell SCHDRAW to load the -1B or -2A Body Object, respectively. Getting back to the 74LSXXX component, notice the Body Object **BODY-PIN hidden** power and ground pins (@) attributes assignment lines (**I1, O2, I16 | @+5, and I8 | @GND**) are not displayed.

Also, the **PARM N/R** fields (**NAME... and REFD**) are replaced with the component name, **74LSXXX**, and reference designator, **U3**.

Connect and Label 74LSXXX Component

Connect pin 2 of the 74LSXXX to U2, pin 3, as shown in figure 1-15. Next, connect a signal wire from 74LSXXX, pin 1, as shown in figure 1-15. Now select **LABEL** and enter:

CLK <Enter>

Place the **CLK** label at the end of the 74LSXXX, pin 1 wire, as shown in figure 1-15.

We are now finished drawing the LOGIC schematic. Please carefully check your work against the completed schematic shown in figure 1-15. Ensure all labels are placed correctly and all signal wires begin/terminate correctly. We will now *quickly* cover some Schematic Editor commands we have not used. Then we will run SCHEMA III post-processor, SCHPOST, on the LOGIC schematic to check for design and drawing errors and produce a bill of materials and other post-processing outputs.

Before proceeding, we will save the LOGIC schematic. Select **FILE**. The **FILE** Command Box lights and **FILE** Secondary Commands are displayed in the lower portion of the Command Window. These **FILE** commands are used to save and load drawing files, redraw a current drawing, print or merge a drawing, invoke the Object Editor, and return to the Manager or DOS.

Select **SAVE**. The Text Window appears with the current drawing name, **LOGIC.P01**. Press **<Enter>**. The LOGIC.P01 schematic page is saved to the current directory/disk. Since LOGIC.P01 already exists, SCHEMA III automatically renames the original LOGIC.P01 file as a back-up drawing named LOGIC.B01.

For tutorial purposes a brief explanation of the other primary and secondary commands used in SCHDRAW will follow.

FILE

The **FILE-LOAD** command is used to load a new drawing file into the Schematic Editor. **LOAD** operation is straightforward - just select **LOAD** and enter the desired file name in the Text Window. If the desired drawing file does not reside in the current directory/disk, be sure to enter the correct directory path to the file before the file name. Enter a *.P* to get a file browse to select the file.

Select **REDRAW**. The **REDRAW** Command Box lights and the cursor disappears while the LOGIC drawing is redrawn from memory. After the redraw operation is complete, the **REDRAW** Command Box extinguishes and the cursor reappears. **REDRAW** is generally used after multiple images are placed on top of each other and one is erased. In this instance, the underlying image(s) may appear to be partially erased. Use **REDRAW** to display the true screen contents if the Drawing Window display becomes degraded.

Select **PRINT**. The **PRINT** Command Box lights and the Text Window appears with **PRN** displayed. If you press **<Enter>** then then the schematic will be printed in foreground mode. *Do not press <Enter>!* If you select the command PRINT again the most recently entered file name, appended with a temporary file extension: **LOGIC.T01**. The text displayed in the Text Window is the **PRINT** operation *destination*. If the destination is a file name (currently LOGIC.T01), the drawing is spooled to disk as a print file and a background print operation is initiated using the DOS PRINT.COM program.

Spooling the current file to disk requires a hard disk system because print files are generally extremely large.

FILE-QUIT is discussed later.

FILE-ED OBJ invokes the Object Editor, as previously discussed.

NOTES

Select **NOTES**. The **NOTES** Command Box lights and a set of Secondary Commands appears in the lower portion of the Command Window. These Secondary Commands (**LINE,CIRCLE,TEXT,MOVE, COPY,** and **ERASE**) are used to add non-schematic lines and text to schematic drawings. The SCHPOST post-processors extract connection and component information from schematics and ignore all **NOTES** information. When making general-purpose drawings that do not require post-processing (such as block diagrams, flow charts, etc.), **NOTES** *or* **DRAW** commands can be used to achieve similar results.

LINE works identically to **DRAW-WIRE** except there are four line types available and circles, arcs, and diagonal lines can be drawn. In addition to thin and fat lines, **LINE** also provides dashed and dotted lines. When drawing orthogonal lines, operation is identical to **WIRE**. To draw circles and arcs, select **CIRCLE**. The Text Window appears.

To draw a circle, enter:

<Enter>

(where **64** is the radius of the circle in pixels). After **<Enter>** is pressed, the Text Window disappears. Position the cursor in the Drawing

Window and press and hold the mouse button or press **<Ins>**. The circle appears and can be placed anywhere in the Drawing Window. To place the circle, release the mouse button or repress **<Ins>**. To draw an arc, enter:

<Enter>

(where **32** is the radius of the arc in pixels, **180** is the arc beginning point (in degrees) and **270** is the arc ending point in degrees.). Place the arc next to the circle.

Now let's draw a diagonal **LINE**. First, select **CONFIG** followed by **DIAG**. Next, select **NOTES-LINE** until the thin line icon is displayed. Now position the cursor in the Drawing Window and draw a line. Notice that the line can be drawn at any angle and length.

TEXT characters can be in upper or lowercase and small, medium or large size type. There are assorted Greek and International characters also available. Font size is selected by selecting the **TEXT** Command Box and toggle to the appropriate character style desired. As with **LABEL**, vertically-oriented text is produced by beginning **TEXT** entries with a backslash character (\), followed by a **<Space>**, and the applicable **TEXT**. As with **LABEL**, **TEXT** entries can be edited using the **EDIT-IMAGE** command discussed later.

All **NOTES** can be undone using **UNDO**.

EDIT

EDIT is used to edit/modify existing images or drawing regions. Select **EDIT**. The **EDIT** Command Box lights and several **EDIT** Secondary Commands appear in the lower portion of the Command Window.

Select the **SELECT** Secondary Command. The **SELECT** Command Box lights. Use **SELECT** to preselect images to be moved, copied, erased, or edited using **EDIT-IMAGE**. Position the cursor over resistor R1 and press **<Ins>** twice or click the mouse button. R1 slowly flashes to indicate it is selected for an **EDIT** operation.

Now select the **MOVE** Secondary Command. The **MOVE** Command Box lights. Next, position the cursor in the Drawing Window and either press **<Ins>** or the mouse button. The resistor jumps to the cursor and can now be moved anywhere on the drawing. Press **<Ins>** or release the mouse button to place the resistor. R1 continues to be selected and slowly flashes. Move the resistor back to its original position.

Now, select the **COPY** Secondary Command. The **COPY** Command Box lights. Position the cursor in the Drawing Window and press **<Ins>** or the mouse button. A copy of the resistor appears at the cursor and can be moved anywhere on the drawing. Notice that a new reference designator was automatically assigned to the copied resistor. Release the mouse button to place the copied resistor. The new copy of the resistor is selected and slowly flashes.

To erase the copied resistor, select **ERASE**. The **ERASE** Command Box lights. Position the cursor in the Drawing Window and either press **<Ins>** twice or click the mouse button. The copied resistor is erased. Now **ERASE** the circle, arc, and diagonal line drawn earlier.

MOVE, COPY, and **ERASE** work with all image types, i.e., wires, lines, labels, objects, arcs, etc.

Labels (and text) are edited using the **EDIT-IMAGE** command. Note that **UNDO** undoes any **MOVE, COPY,** or **ERASE** operation. **UNDO** must be selected *immediately after the applicable command is executed.*

UNDO does not work with **EDIT-WIRE**. As previously mentioned, **EDIT-IMAGE** is used to edit component **DRAW-OBJECT** Parameter Strings, labels and text, and signal wires. We have already used **EDIT-IMAGE** to edit some **DRAW-OBJECT** Parameter Strings. Let's edit the CLK label assigned to pin 1 of the 74LSXXX component. First, select **IMAGE**. Next, position the cursor over CLK and either click the mouse button or press **<Ins>** twice to select the label. The Text Window appears with the CLK label. Now press **<Esc>** to clear the Text Window and enter:

SAMPLE <Enter>

After **<Enter>** is pressed, the CLK label is instantly changed to SAMPLE. Use **EDIT-IMAGE** to change the SAMPLE label back to CLK.

EDIT-WIRE is used to straighten or lengthen signal wires. **EDIT-WIRE** operates on junctions (not *connections*) of two signal wires or at the end point of a single signal wire. We will now use **EDIT-WIRE** to edit two signal wire junctions. First, position the cursor at the signal wire bending point junction near the 74LSXXX, pin 2. Now press the mouse button or **<Ins>** and move the wires to the right as shown in figure 1-16. Release the mouse button or repress **<Ins>** to place the wires. Now use **EDIT-WIRE** again to 'clean-up' the upper signal wire junction as shown in figure 1-17. If the end point of a wire is selected an extra wire is added for use in stair stepping a connection. Working from the two end points of a wire the user can manipulate and alter any wire to a desired configuration. Experiment with this command and get a feel for its operation.

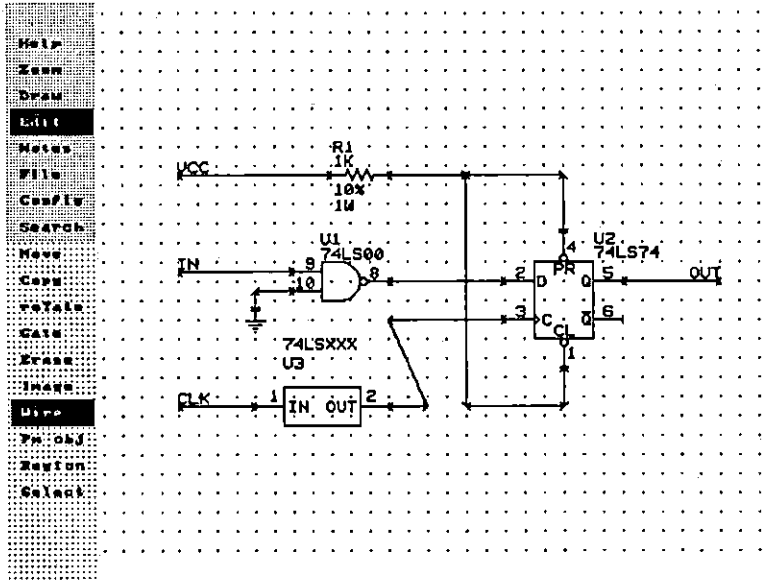


Figure 1-16. Move Wires With EDIT-WIRE

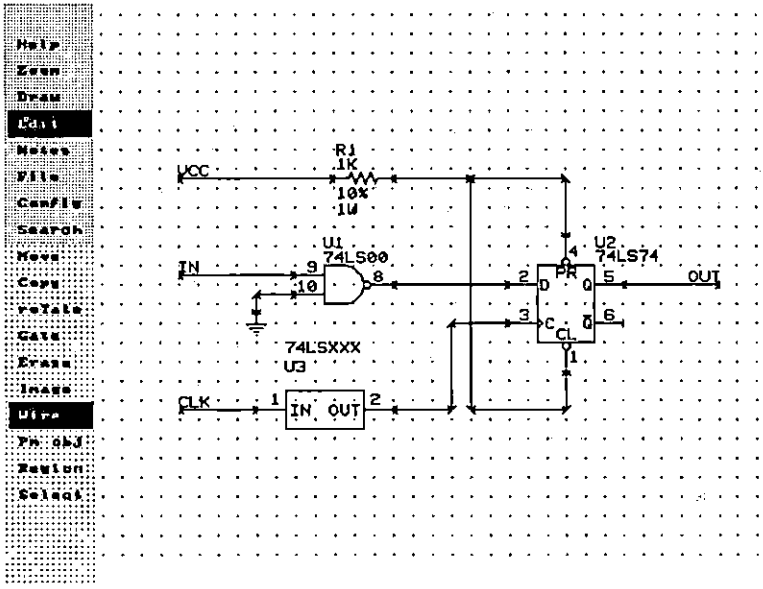


Figure 1-17. Clean-up With EDIT-WIRE

Now let's explore the **REGION** command which allows user-defined areas of a drawing to be edited very quickly. Select **REGION**. The **REGION** Command Box lights and the **EDIT-PM OBJ** and **EDIT-IMAGE** Command Boxes are replaced with the **EDIT-SAVE** and **EDIT-LOAD** Command Boxes, respectively. The **NEWREF** Command Box is added to the Command Window as well. *If the region box already appears, when REGION is selected you can create a new region box.* Pan the drawing until you can see the upper left corner of the LOGIC circuit. Position the cursor to the upper left corner of the circuit. Now either press **<Ins>** or the mouse button and move the mouse or cursor in a diagonal direction toward the bottom right corner of the LOGIC circuit. As it moves, a box is formed around the LOGIC circuit. Continue to move it until the entire LOGIC circuit is within the region box. The screen should now resemble figure 1-18. Now press **<Ins>** or release the mouse button. The area contained within the box is the region that can now be moved, copied, erased, or saved.

Select **NEWREF**. The **NEWREF** Command Box toggles between **ON** and **OFF**. If highlighted, **NEWREF** is **ON**, this allows new reference designators to be generated for all of the parts contained in the region when using **REGION-LOAD** or **REGION-COPY**. If not highlighted, **NEWREF** is **OFF**, this allows the current reference designators to remain the same. New reference designators are *not* generated for all of the parts contained in the region.

Region Load will also allow you to load a piece of text as a region. Select **MOVE**. The Move Command Box lights. Position the cursor in the center of the Drawing Window and press **<Ins>** or press and hold the mouse button. The **REGION** box disappears and the enclosed circuit is selected. Move the region a little then repress **<Ins>** or release the mouse button. The **REGION** box is redrawn, enclosing the region. Next, select **ERASE** (**REGION** Command Box should still be lighted and the **REGION** box should be displayed in the Drawing Window). The **ERASE**

Command Box lights. Position the cursor anywhere in the Drawing Window and press <Ins> twice or click the mouse button. All items within the REGION box are erased.

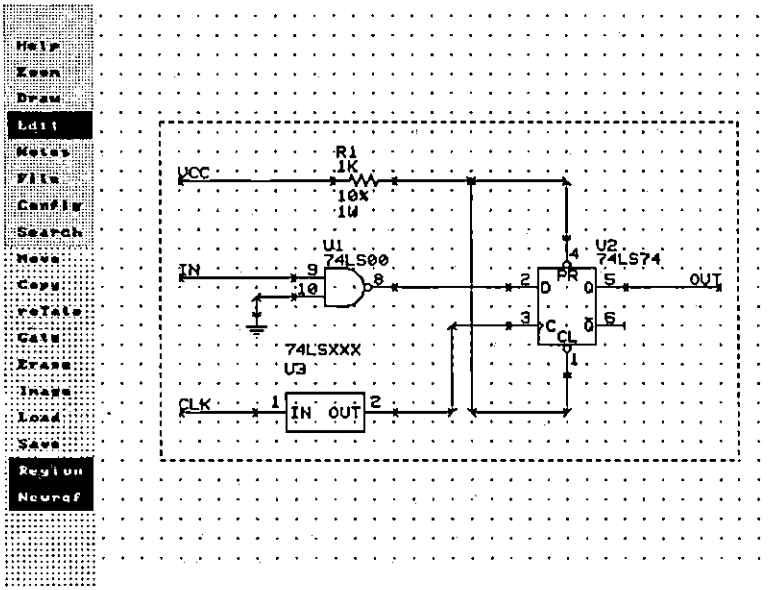


Figure 1-18. REGION

Now select UNDO. After a few seconds (entire screen must be redrawn), the LOGIC circuit is restored. REGION UNDOs restore a drawing to the state it was in just before the region was originally defined. UNDO works with all REGION operations.

REGION-COPY is similar to REGION-MOVE. Select REGION to remove the current region and now reselect REGION to define a new region. Create a new region around a portion of the LOGIC circuit and

select COPY. The COPY Command Box lights. Position the cursor in the Drawing Window and press <Ins> or the mouse button.

A copy of the region appears and can be moved anywhere on the drawing. Repress <Ins> or release the mouse button to place the copied region. Now select UNDO to remove the REGION-COPY.

The REGION-SAVE command is used to save a defined region as a region *file*. When REGION-SAVE is used, the defined region is saved as file REGION.RGN (where REGION is the name of the region and .RGN is the region file extension). The region is the same as a schematic file. If you include an extension ".P01" then a new drawing page is created. This can be useful if you wish to print of plot only one section of the schematic. The REGION-LOAD command is used to merge a previously saved region (REGION.RGN) file or a schematic page (DRAWING.P01) into the current drawing. Note the region being loaded must be at least 2 grid units smaller in both the X and Y coordinates to be loaded into another page. Chapter 3, *SCHDRAW*, contains additional information on REGION-LOAD and REGION-SAVE.

SEARCH

SEARCH is used to locate a particular coordinate, object or text in a drawing and automatically center it in the Drawing Window. Select SEARCH. The Command Box lights and several Secondary Commands appear in the lower portion of the Command Window.

Select the X,Y Secondary Command. The X,Y Command Box lights and the Text Window appears. To find the coordinate, **14.08, 13.01**. Type:

<Enter>

The cursor and the coordinate being searched will be centered automatically in the Drawing Window.

NOTE

Decimal value coordinates are optional and must be between .00 and .15. A digit must be entered before the decimal point (e.g., 0.13).

This is extremely useful with the SCHPOST post-processor. When errors are listed with X,Y coordinates, you can find them quickly using **SEARCH-X,Y.**

Select **NAME.** The Command Box lights and the Text Window appears. To find the object **74LS00,** type:

74LS00 <Enter>

The cursor appears on the object and 74LS00 is centered in the Drawing Window. If you wanted to find all objects beginning with **74LS,** then enter:

74LS* <Enter>

The cursor appears on object 74LS74. Press the left mouse button or the **<Ins>** key and the cursor appears on 74LS00. All 74LS objects will be found. Repeat this until the message, **NO MORE MATCHES,** appears.

The wildcard, **?,** can be used, but a character must reside in that digit for a match to be found. For example, if **74LS?4** is entered, the **74LS74** object will be found in this particular drawing.

NOTE

*SEARCH will only accept 20 characters. If more than 20 are needed, enter 19 characters followed by an *. SEARCH will not find orientations.*

Select **REFD**. The Command Box lights and the Text Window appears. To find a specific reference designator, **R1**, for example, enter:

R1 <Enter>

The cursor will appear on the object and the parameter string will appear in the Text Window to be edited if you choose.

To find all objects with auto-assigned reference designators, enter:

.* <Enter>

in the Text Window.

LABEL and **TEXT** work the same as **NAME**. However, **LABEL** and **TEXT** are upper or lower case sensitive.

<Left Blank>

LESSON 6- USING HIERARCHY

1.5.4.6 Hierarchical Drawings

LOGIC.P01 is a "flat" drawing, each part and wire represents actual parts and wires. The drawing has only one level, "flat". On the other hand, hierarchical drawings contain signals and objects that represent actual wires and parts plus an object that represent another drawing. One schematic drawing can be represented by one Hierarchical Body Object which is a part of a completely different drawing. There is no limit to the number of layers or drawings involved to complete the high-level hierarchical drawing. This design relies on other drawings for additional information to complete the schematic.

NOTE: If you have not completed the previous lessons and you do not have a LOGIC schematic. One has been provided for you. In the \SCHEMA directory there is a LOGIC.ZIP file that contains the schematic. Go to the \SCHEMA directory at the DOS prompt and type:

PKUNZIP LOGIC <Enter>

1.5.4.6.1 Advantages of Hierarchically Structured Drawings

MODULARITY. Hierarchical drawings are modular designs. Modular designs are clearer and are therefore easier to modify. It is simpler to modify a single block of circuits than it is to locate all instances of the circuits in an unstructured design.

TOP DOWN DESIGN. Hierarchy is a feature that is useful in creating schematics from the top down. The design process is simplified by representing familiar circuits as modules or blocks (hierarchical objects) in the schematic.

REPETITION. As stated earlier, the designer need only draw an often repeated circuit once. Then he may use this lower-level drawing several times in the higher level drawing, thus shortening drawing time.

SPACE FOR THE WHOLE CONCEPT. It is easier to fit one concept on a page if you are able to transfer redundant parts off the page.

SHORTENS THE DESIGN PROCESS. Because hierarchical designs are modular, there is no need to draw a circuit more than once which saves time and produces a better, clearer design. Better designs make better products.

1.5.4.6.2 Creating the Tutorial Hierarchical Drawing

We will use this schematic drawing to create a simple hierarchical object. First, return to the Object Editor by selecting FILE followed by ED OBJ.

1.5.4.6.2.1 The Tutorial Hierarchical Body Object

As before, make sure the CONFIG-V GRID, C GRID and WI HDR are lighted. Select the LIBR-NEW command to receive a blank Drawing Window. The completed Hierarchical Body Object is shown in Figure 1-

19. The following paragraphs provide a step-by-step procedure to draw the tutorial hierarchical body object.

Draw Component Shape and Add Pins

Select the **BODY-LINE** Command Box to draw the object shape. Draw a box shape that is similar to the Tutorial Body Object drawn in section 1.5.4.4.8.1. This box should be 5 grid units wide (horizontal) by 6 grid units high (vertical).

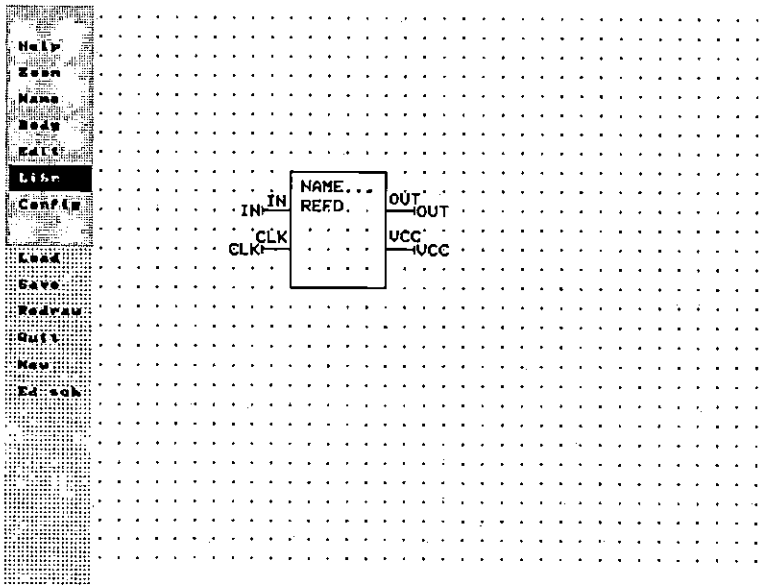


Figure 1-19. Tutorial Hierarchical Body Object

b. Add Pin Definitions

Select **BODY-PIN** until you have the icon with the box on the right of the pin. This is the icon for a pin on the left side of the part. In the Text WIndow type:

<Space>IN <Enter>

Since this part is hierarchical it's pin definitions are net names not pin numbers. The **<Space>** is an attribute that the Object Editor will not think that the 'I' in "IN" is a pin attribute. Place this pin in the upper left half of the part.

Repeat for **CLK** and place it below **IN**.

Next select the **BODY-PIN** command until the icon displays the box to the left of the pin.

Place **OUT** to the right of **IN** and **VCC** to the right of **CLK**.

Define Parameters and Their Locations

Select the **BODY-PARM** command. The **PARM** Command Box lights with the N icon, for the name parameter. Enter:

NAME... <Enter>

Now place the **NAME...** inside the top portion of the box. Reselect the **PARM** Command Box, advance from **N** to **R** (the reference designator parameter). Now enter:

REFD <Enter>

Place **REFD** underneath **NAME...** inside the box.

NOTE

The pin definitions on hierarchical objects must match the net labels on the schematic of the hierarchical object. When the schematic is flattened, the signals going into the pins of the hierarchical object are actually going into the nets on the corresponding hierarchical schematic.

Save the Hierarchical Body Object

After the drawing is complete, select the **LIBR-SAVE** command. Enter:

#HLOG-1A <Enter>

NOTE

Body Object Names for all hierarchical objects must begin with #H. When designing schematics for the purpose of PCB Layout, hierarchical objects/drawings cannot be used repetitively. See Chapter 7, SCHPOST.

1.5.4.6.2.2 The Tutorial Hierarchical Name Object

Now we need to create a Name Object for the Hierarchical Body Object. Select the **LIBR-NEW** command (make sure you have saved the #HLOG-1A Body Object first). Figure 1-20 shows the completed Tutorial Hierarchical Name Object. The following paragraphs provide step-by-step procedures.

a. Enter a Brief Description of the Component

Select the NAME-DESC1 Command Box. Enter the following description of the object in the Text Window:

HIERARCHICAL OBJECT <Enter>

Place this description anywhere in the Drawing Window, as shown in figure 1-20.

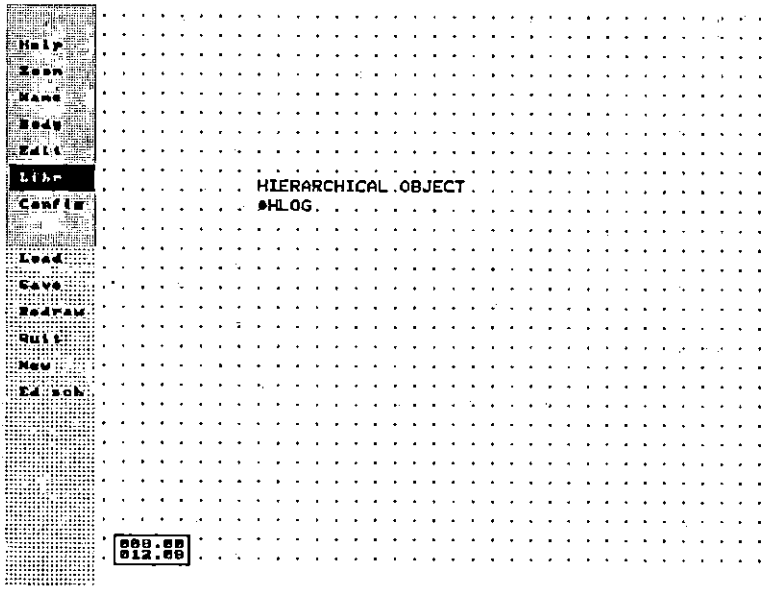


Figure 1-20. Tutorial Hierarchical Name Object

b. Link the #HLOG-1A Body Object to this Name Object

Select the **NAME-BODY** command. Enter the associated Body Object prefix name:

#HLOG <Enter>

The name is limited to 5 characters including the #H (Do not include the orientation/gate section suffix.). Now place **#HLOG** below the **DESC 1** field.

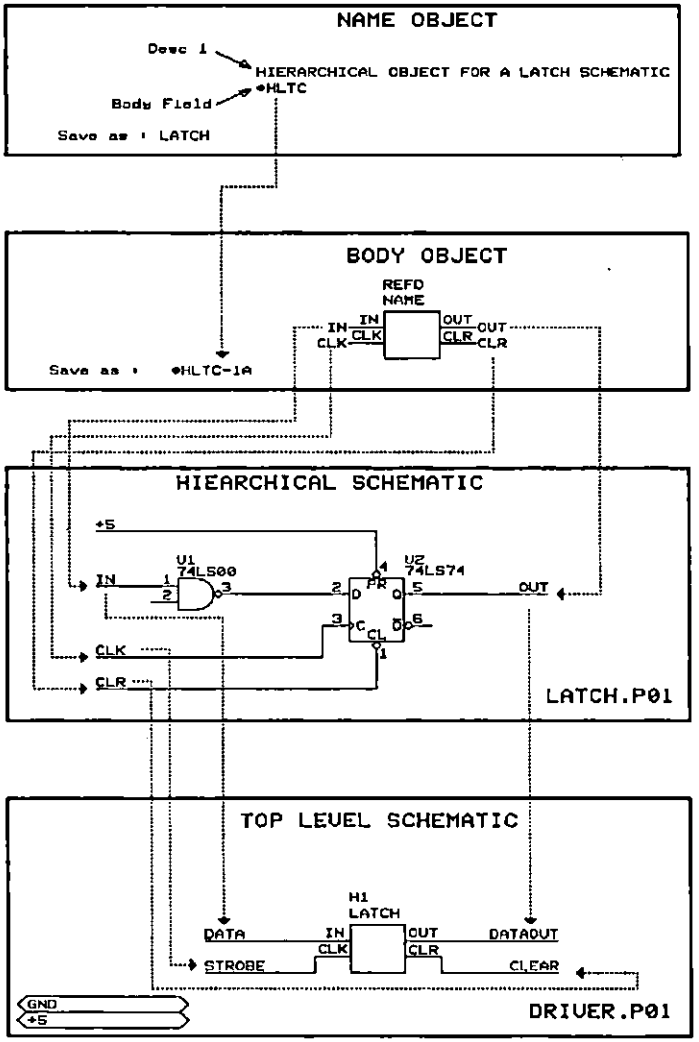
Save the Name Object

We have completed the Name Object and need to save this Name Object to correspond with the schematic page of the same name. Select the **LIBR-SAVE** command and enter:

LOGIC <Enter>

NOTE

The Name Object name for all hierarchical objects must have a corresponding hierarchical schematic of the same name, that shows the circuit of the hierarchical object.


NAME OBJECT
Desc 1
HIERARCHICAL OBJECT FOR A LATCH SCHEMATIC
Body Field
*HLTC
Save as : LATCH
BODY OBJECT
REFD
NAME
IN
CLK
OUT
CLR
Save as : *HLTC-1A
HIEARCHICAL SCHEMATIC
+5
U1
74LS00
U2
74LS74
IN
CLK
CLR
OUT
LATCH.P01
TOP LEVEL SCHEMATIC
H1
LATCH
DATA
STROBE
DATAOUT
CLEAR
GND
+5
DRIVER.P01

1.5.4.6.2.3 The Tutorial Hierarchical Drawing

Now we are ready to draw the top level drawing representing the lower level LOGIC schematic. Select the **ED-SCH** to get out of the Object Editor and in to the Schematic Editor. Next, Select **FILE-QUIT** to exit the system. The SCHEMA Program Manager appears. Highlight **NEW_PAGE** to choose a drawing file and press **<Enter>** or a mouse button. Use the **<Arrow>** keys to highlight the **BPAGE2.P99** page size. Press **<Enter>**. We will name the top level drawing **4LATCH**. Enter:

4LATCH.P01 <Enter>

Press **<Y>** to create the configuration file. *Make sure that LOGIC.LBR is configured in 4LATCH.CFG.* Save the configuration. Select **QUIT** and press **<Enter>** or a mouse button and the Manager returns with **4LATCH** as the current **FILE**. At this point you can select **SCHDRAW** and a blank Drawing Window will appear on the screen. Figure 1-21 shows the completed schematic.

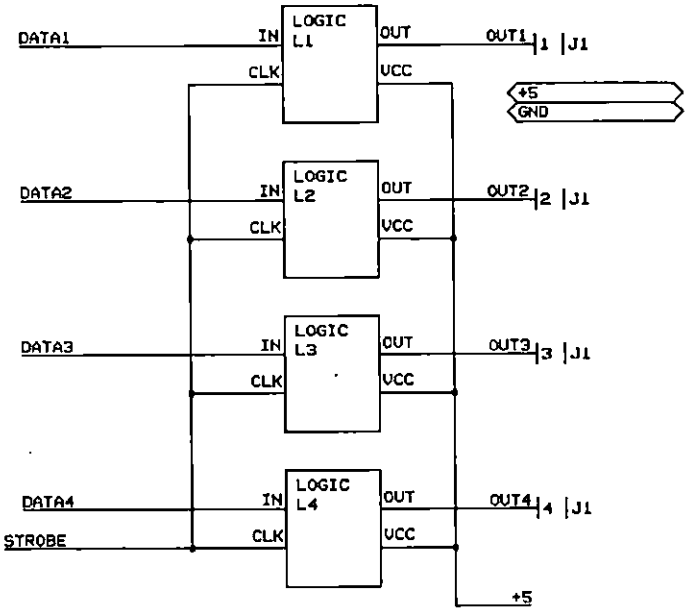


Figure 1-21. Tutorial Hierarchical Drawing

a. CONFIG

To simplify drawing wires and placing labels, we will use the **CONFIG-RPT** command to repeat the same function several times. Select **CONFIG-RPT X**. Press **<Esc>** to clear the entry and enter:

0 <Enter>

The **RPT Y** Command Box lights. **<Backspace>** over -0016 and enter:

128 <Enter>

NOTE

The value entered equals 16 pixels multiplied by the number of Visual Grid units apart.

Add the Hierarchical Object and Assign Reference Designators

Select **DRAW-OBJECT**. Enter:

LOGIC,L1

Position the Object using the **<Ins>** key or the left mouse button centered at the top of the Drawing Window.

We want to have 4 schematic drawing representations for this top level hierarchical drawing. Select the **REPEAT** command box. A copy appears and is placed 2 vertical grid units below the original Object. Press **REPEAT** 2 more times giving a total of 4 Objects. The completed drawing is shown in figure 1-21.

Now change the reference designators by selecting the **EDIT-IMAGE**. Position the cursor over the automatically assigned reference designator

inside the second object and press either the **<Ins>** twice or the left mouse button. The **DRAW-OBJECT** Parameter String appears with the automatically assigned reference designator. **<Backspace>** over **.AAB01** (reference designator number) and type **L2** after the **LOGIC** entry. Enter the following:

LOGIC,L2 <Enter>

Proceed to change the next two reference designators to **L3** and **L4** respectively using the same procedures as above. See figure 1-21 for exact placement.

NOTE: When the schematic is post-processed these reference designators will disappear from the netlist. They will be replaced with Hxxxx prefices. If you wish to see the reference designator path of your hierarchical parts then refer to the **LOGIC.HSF** file, which is a key to understanding which hierarchical prefix refers to which hierarchical object.

Label/Connect the Wires

Now that we have established the X and Y values, we can place the wires and labels the same way we created three more objects. Select **DRAW-WIRE** and draw a wire about 12 visual grid units to the left of the **LOGIC,L1 IN** pin. Then, select the **REPEAT** Command Box 3 times. This will automatically create a wire extending from the **IN** pin on the remaining 3 circuits.

Select **LABEL** with the small square on the top left end of the line in the icon box. Enter:

DATA1 <Enter>

Place the label on top left of the wire as shown in figure 1-21. Select the **REPEAT** Command Box 3 times to place **DATA2, DATA3** and **DATA4** on their respective wires. *The number following DATA will be incremented automatically.*

Select **DRAW-WIRE**. Draw a wire extending 6 visual grid units to the right of the **LOGIC,L1 OUT** pin. Select **REPEAT** 3 times.

Select **LABEL** and toggle until the small square on the top right end of the line in the icon box appears.

Enter:

OUT1 <Enter>

Place **OUT1** on top of the right wire. Select **REPEAT** 3 times to place **OUT2, OUT3** and **OUT4** on their respective wires.

Next, connect the 4 **VCC** pins. Select **DRAW-WIRE**. Draw a wire 3 grid units to the right of the **VCC** labeled pin on the **LOGIC,L1** object. Select **REPEAT** 3 times. Draw a vertical wire connecting all the **VCC** wires and continue 2 grid units below the bottom **VCC** labeled pin header. Draw a wire to the right 4 grid units from that point. See figure 1-21 for the completed drawing.

Select **LABEL** with the icon box on the top right of the line. Enter:

+5 <Enter>

Place **+5** on the top right end of the wire. Next, select **DRAW-DOT**. Press **<Ins>** or the left mouse button to place a dot at the intersection of the second, third and fourth **VCC** wires that intersect with the connector wire.

Now do the same process for the left side of the object to connect the CLK pins. Select **DRAW-WIRE**. Draw a wire extending 3 grid units to the left of the **LOGIC,L1 CLK** pin. Select **REPEAT** to create a wire extending from the CLK pin on the remaining 3 objects.

Select **DRAW-WIRE** to draw a vertical wire connecting the four **CLK** wires. Next, draw a wire left 10 grid units from the bottom **CLK** wire. Select **LABEL** with the icon box on the top left. Enter:

STROBE <Enter>

Place **STROBE** on the left end of the bottom **CLK** line. See figure 1-21 for the completed drawing.

Select **DRAW-DOT** to place dots as done before on the second, third and fourth wire intersections.

Add the Connector Object

Select **DRAW-OBJECT** and enter:

^ ^CONN\2,J1,1,,,001-0001-001 <Enter>

This Parameter String retrieves connector object **^ ^CONN** with an assigned reference designator of **J1**. Also, pin **1** is assigned as the connector pin number and **001-0001-001** the part number. Connect **^ ^CONN** to the end of the wire labeled **OUT1**.

Select **REPEAT** three times to attach 3 more connectors to **OUT2, OUT3** and **OUT4.**

To complete the schematic, select **EDIT-IMAGE**. Position the cursor on the **1** of the second connector you added. Press the **<Ins>** or the left mouse button. Use the **<Left Arrow>** key to cursor to the period preceding the reference designator, **.AAE01,1**. Use the **<Del>** or the **<Backspace>** key to delete the reference designator and pin number. Type J1,2 in its place and enter:

^ ^CONN\2,J1,2,,,001-0001-001 <Enter>

Using **EDIT-IMAGE**, also replace the reference designator and the **1** on the third and fourth connectors with **J1,3** and **J1,4** respectively. See figure 1-21 for the completed drawing.

e. Add the Global Nets

Nets that exist at different hierarchical levels, without being passed down through a hierarchy, are called **Global Nets**. Global Nets must be designated as such at the top level of the hierarchy. We will be designating **+5** and **GND** as **Global Nets** for this tutorial. See specific rules for **Global Nets** in Chapter 7, *SCHPOST*.

Select **DRAW-OBJECT** and enter:

GLOBAL,,+5 <Enter>

Give no reference designator to the GLOBAL objects. Type two commas consecutively to nullify any reference designator.

Use the left mouse button or **<Ins>** key to place the object anywhere to the right of the schematic.

Next, add a ground global net by entering the following in the Text Window:

GLOBAL,,GND <Enter>

Place the ground symbol below the **+5** object. For the specific placement see figure 1-21.

Save the Completed Drawing.

Finally, save the drawing by selecting **FILE-SAVE.** The Text Window will display:

4LATCH.P01

Press **<Enter>** to save the file.

NOTE

The schematic page that represents the hierarchy must be in the same sub-directory as the the library that contains the hierarchical object (it could also be in the current directory).

1.5.4.6.3 Using the PUSH\POP Feature

First let's define what the terms **PUSH** and **POP** represent. **PUSH** will push a new file onto the stack of available schematics. **POP** will let you select a file from the stack of available schematics and edit it.

Make sure that you have the file **4LATCH.P01** loaded. Select **FILE-PUSH** and place the cursor over one of the hierarchical objects created for this schematic. Press the left button and notice that the text window appears with the file **LOGIC** displayed. Press **<Enter>** and the **LOGIC.P01** schematic is loaded into the Schematic Editor.

Next, notice that the number next to the **POP** command says **'02'**. This means that there are two files on the STACK: **4LATCH** and **LOGIC**. Continue to select **FILE-POP** until the number is **'01'**. This the

top level of the hierarchy and the text window displays the file 4LATCH. Press <Enter> and the 4LATCH schematic will be re-loaded.

Text files or non-electrical schematics can be loaded by assigning PARM 4 with the filename. While still in the 4LATCH schematic, select DRAW-OBJECT and type:

DOCUMENT,,NOTES,,,TEST.TXT <Enter> .

Manufacturing
Notes

Place the document part some where on the schematic. Now select FILE-PUSH and click the mouse on the DOCUMENT part. The filename TEST.TXT will appear in the text window. Press <Enter> and the text file will be loaded. Use POP to return to 4LATCH. Select EDIT-IMAGE and select the DOCUMENT part. Change TEST.TXT to TEST.P01 and press <Enter> to complete the change. Use PUSH again and a non-electrical schematic will be loaded. Text and non-electrical schematics can be assigned to any part. For example, if you have a PAL in your schematic and you wish to have it carry a schematic that demonstrates the glue logic within it, then you would add the filename to PARM 4 as in the document example. Or, if you wanted to display the logic compiler input file you could place this filename in PARM4 .

<Left Blank>

LESSON 7- Using SCHPOST

This completes the tutorial drawing exercise. We will now exit SCHDRAW and invoke SCHPOST to run SCHEMA III post-processing software on the completed 4LATCH schematic. Select **FILE-QUIT** and follow the prompt by entering **<Enter>**. The Manager is displayed. Now, position the cursor over **SCHPOST**. The Manager COMMAND text line displays:

SCHPOST 4LATCH

Press **<Enter>** or a mouse button to invoke SCHPOST with the 4LATCH schematic. The Manager menu is replaced with the SCHPOST menu, shown in figure 1-22.

1.5.4.7 SCHPOST

SCHPOST provides the post-processing function for SCHEMA III. SCHPOST uses schematic drawings produced in SCHDRAW as input to create up to seven unique post-processor outputs.

The SCHPOST menu displays a list of options. These options select the applicable post-processor(s) to run. We will run all the post-processors, option A, on the 4LATCH schematic, so select **ALL SUB POST-PROCESSORS** and press **<Enter>**.

For this tutorial exercise, all output is sent to the screen except the net and pin list outputs, which are sent to the current directory/disk under file name 4LATCH.NET and 4LATCH.NPL, respectively. When using SCHEMA-PCB, the pin list and SCHEMA-PCB pin list outputs are not generated by the A (all) option. *The SCHEMA-PCB pin list output is only generated when the S option is chosen*, which is created under a file name

with the extension .SCH in the \PCB\FILES directory.

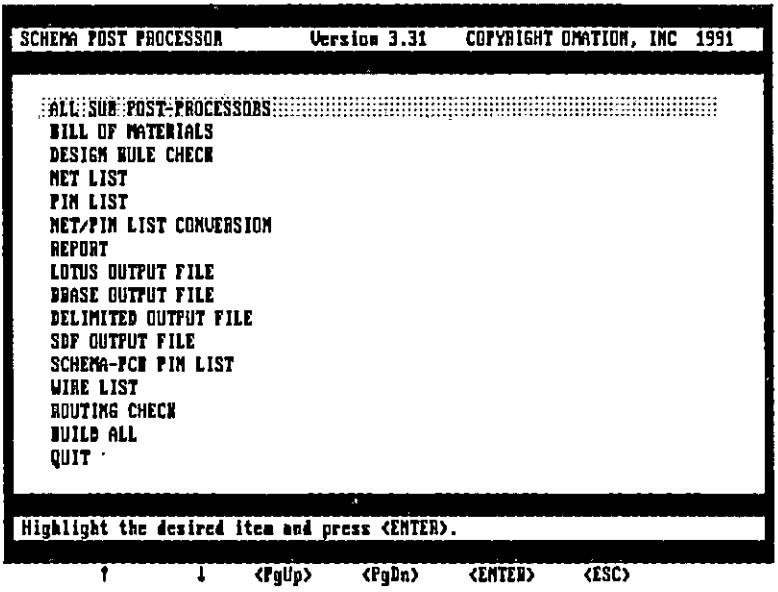


Figure 1-22. SCHPOST Menu

NOTE

The BOM output is 120-columns wide. Thus, the BOM screen display appears wrapped. All other post-processor outputs are 80-columns wide.

Simultaneously press the **<Ctrl>** and **<S>** keys to stop the post-processor outputs as they scroll on the screen. Press any key to continue scrolling. Outputs are displayed in the following order: Routing, Reports, Design Rule Check, Wire List, and Bill of Materials.

Another way to run SCHPOST is to enter:

SCHPOST <DRAWING> <OPTION LETTER>

where **DRAWING** is the name of the drawing and **OPTION LETTER** is the letter corresponding to the post processor. The options are discussed in more detail in Chapter 7, *SCHPOST*. If you want to create a hardcopy of the output, rerun SCHPOST by entering:

SCHPOST 4LATCH A >PRN <Enter>

in the Manager COMMAND text line. This command line runs SCHPOST with the 4LATCH schematic as input, selects all post-processors ('A' except SCHEMA-PCB Pinlist), and redirects output to your printer (**>PRN**).

Compare the SCHPOST outputs with your 4LATCH schematic. Figures 1-23 through 1-26 contain the 4LATCH schematic SCHPOST outputs. A brief description of each SCHPOST output follows.

a. Routing. The Routing post-processor lists various low-level connection errors including unconnected signal wires, incorrect connection junctions, and floating labels. The Routing post-processor output is not generated if there are no routing errors. Thus, if you drew the 4LATCH schematic correctly, you will *not* have a Routing post-processor output. NETS WITH MORE THAN ONE LABEL and REFERENCE DESIGNATORS USED TWO OR MORE TIMES are two routing errors than are also checked . If you receive a Routing output, examine your 4LATCH schematic carefully for the aforementioned errors.

b. Reports. The Usage Reports post-processor creates two reports (figure 1-23) to aid general-purpose design documentation. One report lists all labels and the drawing page(s) on which each label is used. The

other report lists each reference designator used followed by the applicable component name, gate section(s) and page(s) on which used.

c. Bill of Materials. The Bill of Materials (BOM) post-processor provides a standard component usage report in a user-configurable output format - see figure 1-24. Each component's type, reference designator, quantity used, description, etc. can be listed. BOM content and format is completely user-defined. BOM files can even be created to be used as inputs to other programs. Currently, the BOM output is configured to list an item number, followed by quantity used, part number, and component description. Note that the part number and component description data is taken from each component's Name Object **DESC 1** and **DESC 2** fields.

d. Design Rule Check. The Design Rule Check (DRC) post-processor runs six design checks on a drawing and reports the results. The six checks are: floating inputs, nets with no driving source, nets with multiple driving sources, nets with a single pin, and pins with more than one net. These checks catch most common design errors.

REPORT REF. DES. Tue Mar 19 16:04:24 1991 PAGE 1
4LATCH SCHEMATIC 150-0001-001 REV 1A 7/06/88

H1000/R1	DRES	A/01
H1000/U1	74LS00	C/01
H1000/U2	74LS74	A/01
H1000/U3	74LSXXX	A/01
H1001/R1	DRES	A/01
H1001/U1	74LS00	C/01
H1001/U2	74LS74	A/01
H1001/U3	74LSXXX	A/01
H1002/R1	DRES	A/01
H1002/U1	74LS00	C/01
H1002/U2	74LS74	A/01
H1002/U3	74LSXXX	A/01
H1003/R1	DRES	A/01
H1003/U1	74LS00	C/01
H1003/U2	74LS74	A/01
H1003/U3	74LSXXX	A/01
J1	^ ^CONN	A/01,A/01,A/01,A/01

```
---------------------------------------------------------------------------
 REPORT LABEL          Tue Mar 19 16:04:25 1991      PAGE 1
 4LATCH SCHEMATIC  150-0001-001 REV 1A  7/06/88
---------------------------------------------------------------------------


+5              01
DATA1           01
DATA2           01
DATA3           01
DATA4           01
GND             01
OUT1            01
OUT2            01
OUT3            01
OUT4            01
STROBE          01
```

Figure 1-23. 4LATCH Usage Reports Post-Processor Output

BILL OF MATERIALS XYZ COMPUTER CORP. 4LATCH
150-0001-001 REV 1A 1

ITEM	QUAN.	PART NUMBER	DESCRIPTION	REF. DES.
1	1	001-0001-001	CONNECTOR	J1
2	4	050-0010-001	74LSXXX - QUAD NOTHING GATE	H1000/U3 H1001/U3 H1002/U3 H1003/U3
3	4	050-1000-025	RESISTOR,1K,	H1000/R1 H1001/R1 H1002/R1 H1003/R1
4	4	300-0000-001	74LS00 - LS TTL QUAD 2 IN NAND	H1000/U1 H1001/U1 H1002/U1 H1003/U1
5	4	300-0074-001	74LS74 - LS TTL DUAL D FLIP-FLOP	H1000/U2 H1001/U2 H1002/U2 H1003/U2

Figure 1-24. 4LATCH Bill of Materials Post-Processor Output

```
------------------------------------------------------------------------
DESIGN RULE CHECK       Tue Mar 19 16:06:09 1991      PAGE 1
4LATCH SCHEMATIC  150-0001-001 REV 1A  7/06/88
------------------------------------------------------------------------


FLOATING INPUTS:
  NONE FOUND

NETS WITH NO DRIVING SOURCE:
%STROBE         H1000/U3-I1/01   H1001/U3-I1/01   H1002/U3-I1/01
                  H1003/U3-I1/01

NETS WITH MULTIPLE DRIVING SOURCES:
  NONE FOUND

NETS WITH A SINGLE PIN:
%DATA1          H1000/U1-I9/01
%DATA2          H1001/U1-I9/01
%DATA3          H1002/U1-I9/01
%DATA4          H1003/U1-I9/01

PIN WITH MULTIPLE NETS:
  NONE FOUND
```

Figure 1-25. 4LATCH DRC Post-Processor Output

e. Wire List. The Wire List post-processor creates a general-purpose connectivity list shown in figure 1-26. The list consists of a series of nets. Each net consists of an item number followed by a label and the list of components to which it connects. If a label is unassigned, as shown in items 3 - 6, SCHPOST automatically assigns a label beginning with a dollar sign character ($) followed by the first component's reference designator, pin attribute (**BODY-DF PI** assignment), and applicable schematic page number. Notice that SCHPOST automatically lists the Vcc (+5) and ground (GND) nets taken from the component's **BODY-DF PI** power and ground assignment lines. A wire list is generally required for prototype construction and/or to aid design documentation.

f. Net List. The Net List post-processor creates a computer-readable connectivity file. This file contains component type data (in addition to the same information as the wire list) and is formatted to be read by another computer program. Generally the net list file is output to another computer system such as a PCB layout system, logic simulator, silicon compiler, or other back-end processor. As mentioned above, the 4LATCH net list is in the file 4LATCH.NET located in the current directory/disk.

g. Pin List. The Pin List post-processor is similar to the Net List post-processor except it contains comprehensive component information in addition to net information. The 4LATCH pin list is in the file 4LATCH.NPL located in the current directory/disk. The SCHEMA-PCB pin list was not generated if using the option for all post-processors with SCHEMA-PCB.

Chapter 7, *SCHPOST*, contains additional information and comprehensive descriptions of each SCHPOST post-processor option and output.

When SCHPOST finishes, the Manager menu is displayed. This completes the SCHPOST tutorial. To print or plot the 4LATCH schematic, proceed to the next paragraph.

```
-------------------------------------------------------------------------
 WIRE LIST              Tue Mar 19 16:06:35 1991      PAGE 1
 4LATCH SCHEMATIC  150-0001-001 REV 1A  7/06/88
-------------------------------------------------------------------------

1    +5        H1000/R1-L    H1000/U1-14   H1000/U2-14
               H1000/U3-16   H1001/R1-L    H1001/U1-14
               H1001/U2-14   H1001/U3-16   H1002/R1-L
               H1002/U1-14   H1002/U2-14   H1002/U3-16
               H1003/R1-L    H1003/U1-14   H1003/U2-14
               H1003/U3-16

2    DATA1     H1000/U1-9

3    DATA2     H1001/U1-9

4    DATA3     H1002/U1-9

5    DATA4     H1003/U1-9

6    GND       H1000/U1-10   H1000/U1-7    H1000/U2-7
               H1000/U3-8    H1001/U1-10   H1001/U1-7
               H1001/U2-7    H1001/U3-8    H1002/U1-10
               H1002/U1-7    H1002/U2-7    H1002/U3-8
               H1003/U1-10   H1003/U1-7    H1003/U2-7
               H1003/U3-8
```

7	H1000/01036020	H1000/R1-R	H1000/U2-1	H1000/U2-4
8	H1000/01036026	H1000/U1-8	H1000/U2-2	
9	H1000/01036032	H1000/U2-3	H1000/U3-2	
10	H1001/01036020	H1001/R1-R	H1001/U2-1	H1001/U2-4
11	H1001/01036026	H1001/U1-8	H1001/U2-2	
12	H1001/01036032	H1001/U2-3	H1001/U3-2	
13	H1002/01036020	H1002/R1-R	H1002/U2-1	H1002/U2-4
14	H1002/01036026	H1002/U1-8	H1002/U2-2	
15	H1002/01036032	H1002/U2-3	H1002/U3-2	
16	H1003/01036020	H1003/R1-R	H1003/U2-1	H1003/U2-4
17	H1003/01036026	H1003/U1-8	H1003/U2-2	
18	H1003/01036032	H1003/U2-3	H1003/U3-2	
19	OUT1	H1000/U2-5	J1-1	

```
---------------------------------------------------------------------
 WIRE LIST          Tue Mar 19 16:06:36 1991      PAGE 2
 4LATCH SCHEMATIC  150-0001-001 REV 1A  7/06/88
---------------------------------------------------------------------

20   OUT2          H1001/U2-5    J1-2

21   OUT3          H1002/U2-5    J1-3

22   OUT4          H1003/U2-5    J1-4

23   STROBE        H1000/U3-1    H1001/U3-1    H1002/U3-1
                   H1003/U3-1
```

Figure 1-26. 4LATCH Wire List Post-Processor Output

LESSON 8- Using SCHPLOT & SCHPRINT

1.5.4.8 SCHPLOT/SCHPRINT

SCHPLOT and SCHPRINT provide plotting and printing, respectively, for SCHEMA III drawings. Both SCHPLOT and SCHPRINT operate in foreground. Thus, no other operations can be done while printing or plotting from these programs. SCHPLOT is the only SCHEMA III program that produces plots while printing can be done from either SCHPRINT or the SCHDRAW Schematic Editor (using the FILE-PRINT command).

When using FILE-PRINT in SCHDRAW, you have the option to either print in foreground or spool the print to disk and perform printing as a background task. If you have a hard disk system, background printing is an excellent feature, since printing will not tie-up the system and you will be free to do other computer tasks.

1.5.4.8.1 Invoking the SCHPLOT Tutorial

To invoke SCHPLOT, position the cursor over **SCHPLOT** and press **<Enter>**. The Manager COMMAND text line displays:

SCHPLOT 4LATCH

Press **<Enter>** to invoke SCHPLOT with the 4LATCH schematic.

The plot should finish in about 10 minutes. When the plot finishes, control is returned to the Manager.

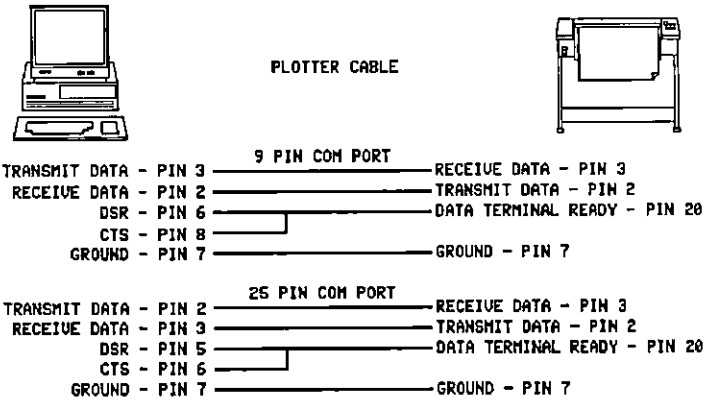
1.5.4.8.2 Invoking the SCHPRINT Tutorial

To invoke SCHPRINT, position the cursor over **SCHPRINT** and press **<Enter>**.

The Manager COMMAND text line displays:

SCHPRINT 4LATCH

Press **<Enter>** to invoke SCHPRINT with the 4LATCH schematic. This prints the tutorial schematic drawing in the normal print mode. The print operation should take about 20-30 minutes, depending on the graphics print speed capability of your printer. There are several print modes available - refer to Chapter 3, *SCHDRAW*, for additional information. After the print finishes, control is returned to the Manager.


PLOTTER CABLE
9 PIN COM PORT
TRANSMIT DATA - PIN 3
RECEIVE DATA - PIN 2
DSR - PIN 6
CTS - PIN 8
GROUND - PIN 7
RECEIVE DATA - PIN 3
TRANSMIT DATA - PIN 2
DATA TERMINAL READY - PIN 20
GROUND - PIN 7
25 PIN COM PORT
TRANSMIT DATA - PIN 2
RECEIVE DATA - PIN 3
DSR - PIN 5
CTS - PIN 6
GROUND - PIN 7
RECEIVE DATA - PIN 3
TRANSMIT DATA - PIN 2
DATA TERMINAL READY - PIN 20
GROUND - PIN 7

LESSON 9- Using SCHLIB and Library Management

1.5.4.9 SCHLIB

The SCHLIB program provides object library file management. A SCHEMA III library file appears to DOS as a single file, but may contain literally *thousands* of individual SCHEMA III objects.

Only SCHEMA III programs have the ability to read and write libraries at the object level. Think of SCHLIB as a set of object library DOS utilities.

SCHLIB performs the following tasks: library creation, library status, copying single (or blocks of) objects from one library to another, and erasing objects. In addition to these functions, SCHLIB also provides a means of creating/editing Name Objects.

1.5.4.9.1 Invoking the SCHLIB Tutorial

To invoke SCHLIB, position the cursor over **SCHLIB**. The Manager COMMAND text line displays:

SCHLIB

Press **<Enter>** or a mouse button to invoke SCHLIB. The Manager menu is replaced with the SCHLIB Main Menu shown in figure 1-27. If you are using a floppy disk-based system, place the Standard Library disk in Drive B.

Before we can use SCHLIB, we must assign one or two libraries. We want to work with two temporary libraries referred to as LIB1 and LIB2. (There is no association between these temporary libraries and the LIBR1 - LIBR9 variables in SCHCFG.). All SCHLIB operations can

then access one or both of these libraries.

Select **LIB1** and press **<Enter>** or a mouse button. The following prompt appears:

ENTER PATH AND FILE FILTER

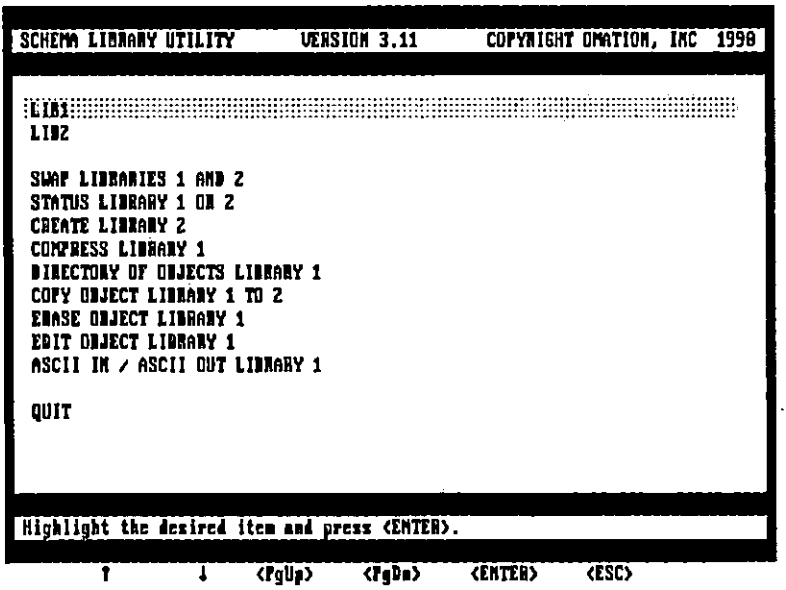

SCHEMA LIBRARY UTILITY VERSION 3.11 COPYRIGHT OMATION, INC 1990
LIB1
LIB2
SWAP LIBRARIES 1 AND 2
STATUS LIBRARY 1 OR 2
CREATE LIBRARY 2
COMPRESS LIBRARY 1
DIRECTORY OF OBJECTS LIBRARY 1
COPY OBJECT LIBRARY 1 TO 2
ERASE OBJECT LIBRARY 1
EDIT OBJECT LIBRARY 1
ASCII IN / ASCII OUT LIBRARY 1
QUIT
Highlight the desired item and press <ENTER>.
↑ ↓ <PgUp> <PgDn> <ENTER> <ESC>

Figure 1-27. SCHLIB Main Menu

TTL1.LBR <Enter>

NOTE

If you have installed the library somewhere other than in the present working directory, the full path must be specified.

Next, enter:

Select **LIB2** and press **<Enter>**. Next, enter:

GRAPH.LBR <Enter>

LIB1 is now set to library TTL1.LBR and LIB2 is set to the library GRAPH.LBR, as displayed on the library assignment line.

Highlight **LIB1** and press **<Enter>** or a mouse button. Next, enter:

***.LBR <Enter>**

This performs a DOS directory search of all files with a .LBR extension in the current directory/disk. This command is a standard DOS directory lookup and can be used for any DOS directory search, but is normally used to display library names. Re-select **TTL1.LBR.**

NOTE: LIB1 & LIB2 have nothing in common with the LIBR1-9 command line in the CONFIG FILES. These are only temporary pointers to different Libraries for SCHLIB to perform object management.

Select **DIRECTORY OF OBJECTS LIBRARY 1** and press **<Enter>**. At the prompt enter:

74LS1* <Enter>

This executes a directory search of TTL1.LBR, looking for all Name Objects beginning with 74LS1. The applicable Name Objects are displayed in the upper portion of the display area. If your printer is installed, enter:

74LS* <Space> \ <Enter>

to output the search string to your printer. (The backslash character (\) outputs the directory listing to the printer, instead of the screen.).

The **ERASE OBJECT LIBRARY 1** command erases objects from the specified library in LIB1. For example: **74LS1* <Enter>** erases all Name Objects beginning with 74LS1 from TTL1.LBR. *Do not try this.*

Select **SWAP LIBRARIES 1 AND 2** and press **<Enter>**. This swaps library assignments. TTL1.LBR is now assigned to LIB2 and GRAPH.LBR is assigned to LIB1. This command is useful when moving objects between libraries.

Select **SWAP LIBRARIES 1 AND 2** press **<Enter>** to swap the libraries back to their original assignments.

Select **COPY OBJECT LIBRARY 1 TO 2** and press **<Enter>**.

Enter:

74LS00 <Enter>

This copies the 74LS00 Name Object from TTL1.LBR (LIB1) to GRAPH.LBR (LIB2). As before, * can be used to copy multiple objects. * by itself would copy all objects from TTL1.LBR to GRAPH.LBR (*Do not do this.*).

Select **STATUS LIBRARY 1 OR 2** and press **<Enter>**. The following prompt appears:

ENTER LIBRARY NUMBER, 1 OR 2

Enter:

1 <Enter>

This displays current LIB1 (TTL1.LBR) status in the display area. As objects are erased or updated, extra space is created in the object libraries. Thus, object libraries should be compressed periodically to remove this extra space and recover valuable disk space. The status display lists the amount of extra library space in the **TOTAL RECOVERABLE BYTES:** field and **NUMBER OF RECOVERABLE DIRECTORY ENTRIES:** field. Then the following prompt appears:

Press any key to continue...

Press any key and the SCHLIB Main Menu returns to the screen. Select **CREATE LIBRARY 2** and press **<Enter>** for a small library.

Enter:

S <Enter>
TEMP.LBR <Enter>

This creates a new library (automatically assigned to LIB2) in the current directory/disk named **TEMP.LBR**. The maximum number of objects that can be placed in this new library is 250, dictated by the S (for small library) entry. Medium (M) libraries can hold about 1000 objects while large (L) libraries hold about 4000 objects.

Once a library is created, the maximum number of objects is fixed. However, a library can be expanded by creating a new, empty, larger library and then copying all objects from the old library to the new, empty, larger library.

To create this library, select **EDIT OBJECT LIBRARY 1** and press **<Enter>**.

The SCHLIB Main Menu is replaced with the Name Object Editor Menu, shown in figure 1-28. This menu allows Name Objects to be created/edited from SCHLIB. (Name Objects are also created/edited in SCHDRAW's Object Editor.).

The Name Object Editor Menu object text fields display the four **DESC** fields and **BODY** field associated with Name Objects. Current values are displayed next to each field (currently empty).

Select **OBJECT** and press **<Enter>**. Enter:

74LS00 <Enter>

This loads a 74LS00 Name Object for editing. The **DESC 1, DESC 2,** and **BODY** fields are filled with values associated with the 74LS00 object.

Select **DESC1** and press **<Enter>**. This moves the current **DESC 1** value to the text line for editing. Enter a different part number such as:

1100-0100-01 <Enter>

The **DESC 1** object text field value changes to reflect this new part number entry.
The **DESC2, DESC3, DESC4** and **BODY** commands work identically to **DESC1**, except they operate on the remaining Name Object fields.

Select **SAVE** and press **<Enter>**. Enter:

TEMPXXX <Enter>

This saves the data assigned to the 74LS00 Name Object as Name Object **TEMPXXX** in the TTL1.LBR library.

Select **NEW** and press **<Enter>**. This clears all five Name Object fields

and is normally used when creating a new Name Object from scratch.

To return to the Manager, select QUIT and press **<Enter>** or a mouse button.

The **ASCII IN / ASCII OUT LIBRARY 1** command allows you to Ascii out Name Objects, alter them and Ascii then into the library assigned to **LIB1.**

The Manager menu is displayed.

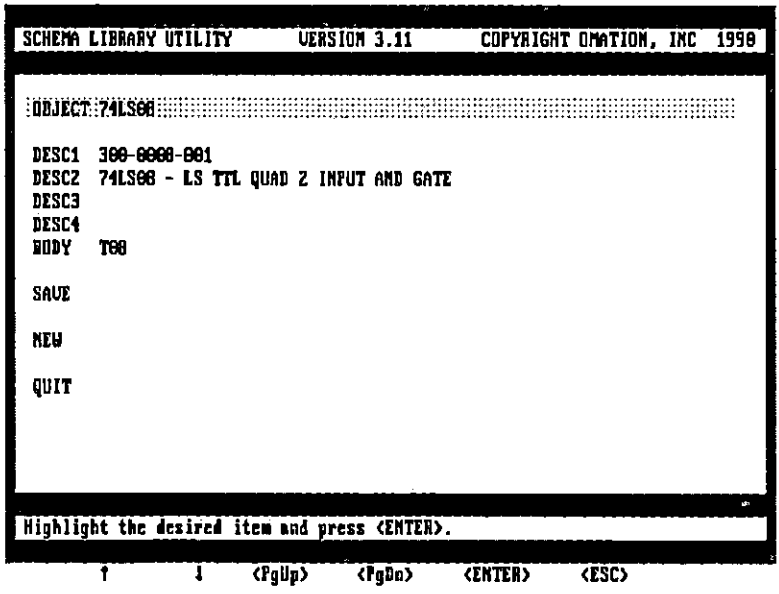


Figure 1-28. SCHLIB Name Object Editor Menu

1.5.4.9.2 Library Philosophy

We suggest that you do not add any new parts to our nine libraries. You should create a User library to store any new parts that are created. This library would be the only library necessary to backup in case of hard disk failure.

1.5.4.9.3 Assign a User Library

After a user library is created it can be assigned to the default position, LIBR1 in the DRAWING.CFG. Where drawing represents the name of the schematic. Thus, when using the Object Editor in SCHDRAW and you save a part and do not specifiy any particular library, it will automatically be saved to your library in the #1 position. Use the SCHCFG program to place your new library into the LIBR1 position of the DRAWING.CFG. Also you may wish to place the new library into the LIBR1 position of the SCHEMA.CFG file, since this is the template file from which all DRAWING.CFG files are created. Then all new projects begun will have your new library in LIBR1

1.5.4.9.4 Archive a Project

Let's say that you have just created a ten page schematic project, PROJECTX.P01 through PROJECTX.P10. However, now you wish to send it to a collegue or store it away , but you don't want to save away all the libraries with it. This is when you should localize your libraries using the program, SCHLOCAL. Use SCHLIB to create a library called PROJECTX.LBR and place this library into the LIBR1 position of the PROJECT.CFG. Then execute the SCHLOCAL on PROJECTX. Thus all the parts used in the schematic are copied to the first library, PROJECTX.LBR. Libraries 2 - 9 can be nullified and only the first library will be used.

1.6 THE SCHEMA-PCB TUTORIAL

The tutorial is based, in part, on the predrawn SCHEMA III schematic drawing named PCB (pages PCB.P01 and PCB.P02). Use SCHDRAW to load and view these files, but *do not edit the files.* Next, invoke SCHPOST with the PCB drawing to generate the pin list file (select the SCHEMA-PCB pinlist, S option). Then, proceed with the tutorial, discussed in Chapter 1 of the *SCHEMA-PCB Layout User's Manual*.

1.7 WHAT NEXT?

This completes the SCHEMA III tutorial. If you followed through the entire tutorial, you should have a good idea of how to use SCHEMA III. Before attempting to actually draw schematics, however, it is strongly recommended that, *at a minimum*, you read Chapter 3, *SCHDRAW.* SCHEMA III utility programs are discussed in Chapter 8, *Utilities.* Please read Chapter 8 before trying to use any of SCHEMA III utility programs.

<Left Blank>

Chapter 2. SCHEMA Program Manager

2.1 INTRODUCTION

The SCHEMA Program Manager, or Manager, provides the interface to SCHEMA III executable programs. While these programs can be invoked directly from DOS, in most cases it is easier, faster, and less error-prone to use the Manager. The Manager lets programs be invoked with a single keypress without the need to remember program or file names. The Manager also allows for quick selection and creation of SCHEMA III drawing and configuration files.

The SCHEMA Manager has been designed to allow the SCHEMA user the capability to add options while maintaining the same Manager. This is to provide a way to coordinate the entire SCHEMA family of products.

In addition, the Manager can support and provide quick access to frequently used DOS commands and other software products the user may have.

2.2 INVOKING THE MANAGER

After installing, DOS must be booted before the Manager is invoked.

To invoke the Manager, make the SCHEMA III directory the current directory by entering:

CD \SCHEMA <Enter>
SCHEMA <ENTER>

from the DOS prompt.
If the AUTOEXEC.BAT, PATH and **OMATION** environment variables have been configured according to the INSTALL instructions, the Manager and SCHEMA executable files may be invoked from any directory. It is not necessary to change directories to the \SCHEMA directory.

When invoking SCHEMA III by entering **SCHEMA** at the prompt, the SCHEMA.BAT, SCHEMA.MGR, and S$.EXE files are used.

Before changing the SCHEMA.BAT or the SCHEMA.MGR file, make a backup copy. This is to insure the you will be able to use the Manager if something should go wrong with the files that you are editing.

2.3 RUNNING THE MANAGER

When the Manager loads, the screen is cleared, and the menu shown in figure 2-1 or 2-1A is displayed.

The Manager menu contains a display area with command columns, COMMAND text line and STATUS block.

The command columns list available Manager commands. The Manager is keyboard or mouse driven. Commands are selected by either using the mouse to highlight the desired command and pressing the left mouse button twice (double click) or using the four **<Arrow>** keys to position the cursor over the desired command and pressing **<Enter>** twice. The left mouse button is equivalent to pressing **<Enter>**. The right mouse button is equivalent to pressing the **<Esc>** key.

To add/edit parameters (path/filename) to the command, press **<Enter>** once. The cursor flashes next to the entry on the COMMAND text line. Make the necessary changes to the command line and press **<Enter>**. The directory will be changed to the appropriate path before execution based on the command execution parameters.

The COMMAND text line also displays user keyboard entries. To execute a command not configured in the Manager, simply begin typing the command as you would at the DOS prompt. The command appears on the command line. Press **<Enter>** to invoke the command.

The STATUS block displays the current settings for the Manager (**PATH, FILE** and **FILTER**). When a command is highlighted, the two HELP lines display any usage messages pertinent to the command selected. These are replaced by error messages when an error has occurred.

A detailed description of each Manager command is contained in paragraph 2.4. Most Manager commands invoke SCHEMA III executable programs.

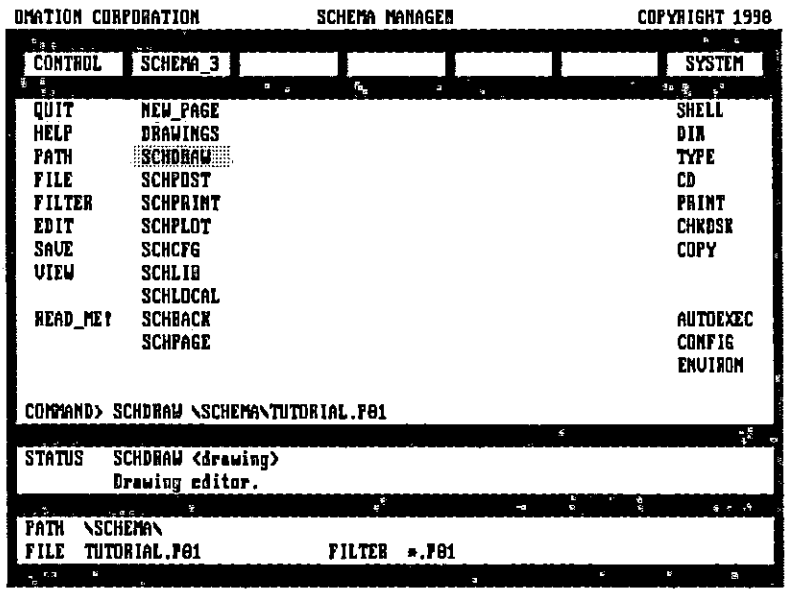


Figure 2-1. The SCHEMA Manager

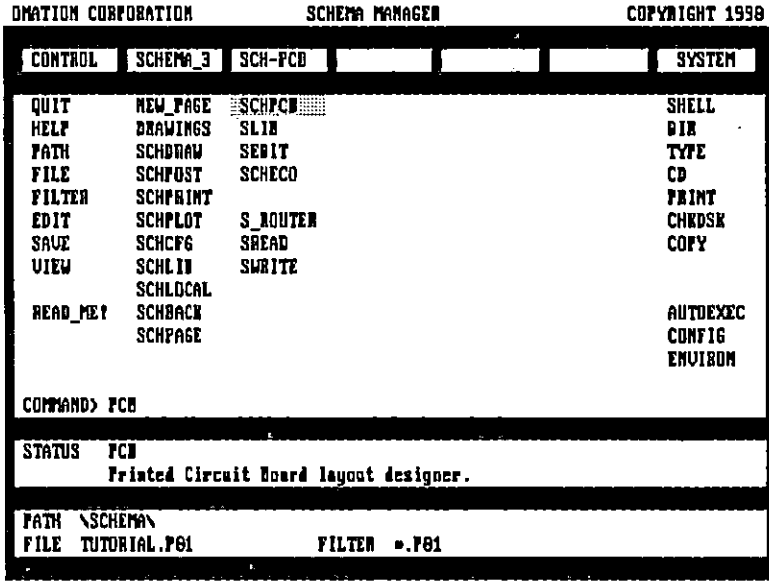


Figure 2-1A. The Manager with the SCHEMA-PCB Interface.

2.4 COMMAND DESCRIPTIONS

This paragraph contains detailed descriptions of Manager commands. The commands are listed in Function key order. Each command is described as follows:

- **Use:** Defines reason for use

- **Operation:** Lists procedure to use command

- **See Also:** Identifies related information, if any

- **Notes:** Identifies special conditions, limitations, etc. associated with command, if applicable.

2.4.1 Manager Control Commands

The following commands are for modifying the Manager environment variables and configuration and to access built-in Manager features.

EDIT

Use: To activate the built-in Manager configuration edit feature.

Operation: Select EDIT and the Manager is placed in Edit Mode. Select the command you wish to edit with the mouse or <Enter>. Make the changes to the command. **BE SURE TO PRESS <ENTER> AFTER EACH LINE ALTERED OR THE CHANGE WILL NOT BE REGISTERED.** When you are finished editing then press the <Up Arrow> to return to the edit screen. Return from Edit by pressing <ESC> once or more. After you have returned to the main manager then use SAVE to record the changees to the manager.

See Also: SAVE command. Customizing the Manager, section X.XX.

Notes: Remember to SAVE the current configuration if the Manager configuration has been edited during the current session and you wish to save the changes or if you wish to preserve the current PATH, FILE, and FILTER settings.

HELP

Use: To activate Manager on-line help.

Operation: Select HELP and the Manager is placed in Help Mode.

While in Help Mode, to view help information for any command in the Manager, select that command. To view help for another command, press <ESC> to return to the listing of commands and select another command.

To quit **Help Mode**, press **<ESC>** again.

Notes: On-line help files are ASCII files with the same name as the corresponding command but with the ext .HLP. Help files for commands added to the Manager by the user can be created with any ASCII Text editor. Help files for SCHEMA commands are provided with SCHEMA III.

FILE

Use: To change the **FILE** (file name) to be used with other SCHEMA III commands.

Operation: Select **FILE** and the command lights. Press **<Enter>** and the cursor flashes next to the COMMAND text line. Enter/Edit the desired file name and press **<Enter>** again. For example, enter:

FILE DRAWING.Pxx

where **DRAWING** is the name of the schematic drawing and **.Pxx** is the page number of the drawing.

Notes: Remember to **SAVE** the current configuration if the Manager configuration has been edited during the current session and you wish to save the changes or if you wish to preserve the current **PATH**, **FILE**, and **FILTER** settings.

FILTER

Use: To change the file and extension that will be used with other SCHEMA III commands.

Operation: Select **FILTER** and the command lights. Press **<Enter>** and the cursor flashes next to the COMMAND text line. Enter the desired filter and press **<Enter>** again. For example, to be able to include all of the drawing pages (blank and created) enter:

FILTER *.P*

where ***.P*** is the desired filter. The **FILTER** in the STATUS block is changed automatically to ***.P***.

See Also: **DRAWINGS**

Notes: This can be used with **DRAWINGS** to display all created drawing pages. Remember to SAVE the current configuration if the Manager configuration has been edited during the current session and you wish to save the changes or if you wish to preserve the current **PATH**, **FILE**, and **FILTER** settings.

PATH

Use: To change the PATH (drive:\directory) that will be used with other SCHEMA III commands.

Operation: Select PATH and the command lights. Press **<Enter>** and the cursor flashes next to the COMMAND text line. Enter the desired path and press **<Enter>** again. For example, enter:

PATH C:\SCHEMA

The **PATH** in the STATUS block is changed automatically to **C:\SCHEMA.**

Notes: Remember to SAVE the current configuration if the Manager configuration has been edited during the current session and you wish to save the changes or if you wish to preserve the current **PATH, FILE,** and **FILTER** settings.

QUIT

Use: To return to DOS.

Operation: Select QUIT and the command lights. Press **<Enter>** to return to DOS.

Notes: Remember to SAVE the current configuration if the Manager configuration has been edited during the current session and you wish to save the changes or if you wish to preserve the current **PATH, FILE,** and **FILTER** settings.

SAVE

Use: To save the Manager configuration to disk.

Operation: Select SAVE and the command lights. Press **<Enter>** and the current settings are saved.

Notes: Remember to SAVE the current configuration if the Manager configuration has been edited during the current session and you wish to save the changes or if you wish to preserve the current PATH, FILE, and FILTER settings.

VIEW

Use: View ASCII text files.

Operation: Press **<Enter>** once with VIEW highlighted to move the cursor to the command line. Type in the name of an ASCII file to view and press **<Enter>** again. Press **<ESC>** to stop viewing and return to the Manager command select window.

Notes: If you have a mouse. The left button is **<Pg Dn>**, the middle button is **<Pg Up>**, and the right button is **<Esc>**.

2.4.2 SCHEMA III Commands

The following commands invoke the SCHEMA III schematic drawing commands and associated project and library maintenance utilities.

DRAWINGS

Use: Browse a directory for and display SCHEMA III drawing files. Also can create new SCHEMA III drawing files when a blank drawing page (*.P99) is selected.

Operation: Select **DRAWINGS** and the command lights. The prompt and text lines display:

DRAWINGS C:\SCHEMA*.P01

where **C:\SCHEMA** is the current PATH and ***.P01** is the current DOS file search string (**FILTER**). Press **<Enter>** and the cursor flashes next to ***.P01**. Two options are available: select an existing drawing or create a new drawing.

1. *To select an existing drawing* press **<Enter>** again to display a list of all drawings in the current directory with page one (**.P01**) extensions. Or, if you want to search for a specific group of files, edit the entry and enter desired file search string. The standard DOS wildcard characters (***** **?**) are supported. For example:

DRAWINGS C:\MEMORY*.P01 <Enter> - Displays all page

1 drawing files from the **MEMORY** directory on the **C** drive.

DRAWINGS C:\MEMORY\POWER.P* <Enter> - Displays all **POWER** drawing files from the **MEMORY** directory on the **C** drive.

DRAWINGS C:\MEMORY\DRAW.P01 <Enter> -
Displays drawing **DRAW.P01** from the **MEMORY** directory on the **C** drive.

After **<Enter>** is pressed the specified file group is displayed in the display area.

To select a drawing, use the mouse or **<Arrow>** keys to position the cursor over the desired file. When the cursor is over the desired file (file name lighted), press the mouse button or **<Enter>**. The FILE prompt displays:

FILE DRAWING

where **DRAWING** is the selected file name. Select the desired program followed by **<Enter>** to invoke the selected drawing with the desired SCHEMA III program.

2. *To start a new drawing for SCHDRAW* select **NEW_PAGE** and press **<Enter>**. The filter is changed to ***.P99** and **DRAWINGS** is automatically invoked. Available blank drawing page files (all drawings in the current directory with a **.P99** file extension) are displayed in the display area.

To select a blank drawing page file, use the mouse or the **<Arrow>** keys to position the cursor over the desired file. When the cursor is over desired file (file name lighted), press the mouse button or **<Enter>**. The prompt line displays:

BLANK PAGE SELECTED. CREATE A NEW DRAWING NAME?
Type yes of no (Y or N), ESC to exit

The cursor flashes in the blank text line. Enter the directory/disk path and name of new drawing with desired page number extension (**.P01** for page 1, **.P02** for page 2, and so on). The file name length (excluding extension) should not exceed 8 characters. SCHEMA III will truncate file names to 8 characters.

After the drawing name and extension are entered, the prompt line displays:

CREATE A NEW CONFIGURATION FILE?
Type yes or no (Y or N), ESC to exit

If you are creating the first page of a new drawing, press **<Y>**. A copy of the SCHEMA.CFG configuration file is made in the current directory or disk and renamed as DRAWING.CFG to match the new drawing name.

If you are not creating the first page of a new drawing (adding another page to an existing drawing set), press **<N>**. Then you can load the new drawing using the **SCHDRAW** command.

See Also: **NEW_PAGE** command

Notes: A drawing or data file must be selected before most SCHEMA III programs can be invoked. Drawings is not limited to browsing SCHEMA drawing files and can be used to browse and select any DOS files for **PATH** and **FILE**.

NEW_PAGE

Use: Browse for and select blank pages (*.P99) from the SCHEMA directory. **NEW_PAGE** calls the **DRAWINGS** command with the blank SCHEMA directory and pages extension as an argument.

Operation: Select **NEW_PAGE** and the command lights. The COMMAND text line displays:

DRAWINGS C:\SCHEMA*.P99

where C:\SCHEMA\ is the path SCHEMA III files were installed. Press <Enter> or the mouse button.

See Also: **DRAWINGS**

Notes: Upon returning, the name of the blank page selected is displayed in the STATUS block, **FILE** parameter. If a new drawing file was created, the new filename and new filename path are displayed in **FILE** and **PATH**, respectively.

SCHBACK

Use: SCHEMA back annotator.

Operation: Select **SCHBACK** and the command lights. The COMMAND text line displays:

SCHBACK DRAWING

where **DRAWING** is the preselected drawing (without extension). A check run option can be specified after **DRAWING**. Refer to Chapter 8, *Utilites*, for additional information on the check run option. Press **<Enter>** to invoke SCHBACK with the selected drawing file (and check run option, if specified).

See Also: *Chapter 8, Utilities*

Notes: A drawing must be selected before SCHBACK can be invoked from the Manager. Refer to the **DRAWINGS** command description for information on selecting drawings from the Manager.

SCHCFG

Use: SCHEMA drawing and default configuration utility.

Operation: Select **SCHCFG** and the command lights. If a drawing has been preselected, the COMMAND text line displays:

SCHCFG DRAWING

where **DRAWING** is the preselected drawing. Press **<Enter>** to invoke SCHCFG with the selected drawing configuration file.

To edit the SCHEMA.CFG file, **<Backspace>** over **DRAWING** and press **<Enter>** (SCHCFG automatically loads the SCHEMA.CFG. default configuration file if invoked with *no drawing name.*).

See Also: Chapter 5, *SCHCFG*

Notes: Refer to the **DRAWINGS** command description for information on selecting drawings from the Manager.

SCHDRAW

Use: Drawing editor for graphical viewing, creation and editing of SCHEMA schematics and library components.

Operation: Select **SCHDRAW** and the command lights. If a drawing has been preselected, the COMMAND text line displays:

SCHDRAW DRAWING

where **DRAWING** is the preselected drawing file (including extension). Press **<Enter>** to invoke SCHDRAW with the selected drawing file. To edit the drawing page number **<Backspace>** over the file name and enter the desired **DRAWING** file name.

See Also: DRAWINGS; Chapter 3, *SCHDRAW*

SCHLIB

Use: Access SCHEMA III libraries to copy, edit, delete, create, erase, and ASCII In/Out library objects.

Operation: Select **SCHLIB** and the command lights. If a drawing has been preselected, the COMMAND text line displays:

SCHLIB

Press **<Enter>** to invoke SCHLIB.

See Also: Chapter 6, *SCHLIB*

Notes: SCHLIB uses SCHEMA III object libraries as input.

SCHLOCAL

Use: Copies all objects used in the specified drawing into the object library assigned to the drawing's configuration file **LIBR1** variable to create a single custom library containing all a schematics components.

Operation: Select **SCHLOCAL** and the command lights. The

COMMAND text line displays:

SCHLOCAL DRAWING

where **DRAWING** is the preselected drawing (without extension). Press **<Enter>** to invoke SCHLOCAL with the selected drawing file. The cursor flashes next to the **DRAWING** file name. If the **DRAWING** is not the desired filename, **<Backspace>** over the name and enter the correct drawing name.

See Also: See Chapter 8, *Utilities*

Notes: A drawing must be selected before SCHLOCAL can be invoked from the Manager. Refer to the **DRAWINGS** command description for information on selecting drawings from the Manager.

SCHPAGE

Use: Create custom or non-standard size blank drawing pages.

Operation: Select **SCHPAGE** and the command lights. Press **<Enter>** and the COMMAND text line displays:

SCHPAGE

Press **<Enter>** and a copyright notice will appear at the top of the screen. Answer the prompts that follow as necessary.

See Also: SCHPAGE, Chapter 8, *Utilities*

Notes: Blank pages created with SCHPAGE should be named with a .P99 extension to distinguish them from non-blank page to the NEW_PAGE, DRAWINGS, and SCHCFG utilities. Custom blank pages can also be created by copying existing blank pages and modifying them with SCHDRAW.

SCHPLOT

Use: Output SCHEMA drawing files to a plotter or vector based device or to a file in HPGL, DMPL, CALCOMP, DXF, or PostScript format as dictated by the "PLOT=" variable in the DRAWING.CFG configuration file.

Operation: Select **SCHPLOT** and the command lights. If a drawing has been preselected and appears as the FILE Manager variable, the COMMAND text line displays:

SCHPLOT DRAWING

where **DRAWING** is the preselected drawing. Press **<Enter>** to invoke SCHPLOT with the selected drawing file (and options, if specified). SCHPLOT options can be specified after **DRAWING**, if desired. Refer to Chapter 4, *SCHPLOT/SCHPRINT*, for information on plot options.

See Also: DRAWINGS; Chapter 4, *SCHPLOT/SCHPRINT*

Notes: Invoking SCHPLOT without a drawing name as an argument will cause SCHPLOT to display a usage message. Refer to the **DRAWINGS** command description for information on selecting drawings from the Manager.

If a DRAWING.CFG configuration file does not exist or is not present, SCHPLOT defaults to the "PLOT=" variable in the SCHEMA.CFG default configuration file.

SCHPOST

Use: Post-process a SCHEMA schematic drawing set to generate net and pin lists, design rule and error checks, component usage reports, Bill of Materials (BOM) listings, and to access format conversions.

Operation: Select **SCHPOST** and the command lights. If a drawing has been preselected, the COMMAND text line displays:

SCHPOST DRAWING

where **DRAWING** is the preselected drawing (less extension). Press **<Enter>** to invoke SCHPOST with the selected drawing file (and options, if specified). SCHPOST options can be specified after **DRAWING**, if desired. Refer to Chapter 7, *SCHPOST*, for information on the post-processing options.

See Also: **DRAWINGS**; Chapter 7, *SCHPOST*

Notes: A drawing must be selected before SCHPOST can be invoked from the Manager. Refer to the **DRAWINGS** command description for information on selecting drawings from the Manager.

SCHPRINT

Use: To print SCHEMA drawing files to rastor devices or to a file in a bit image format as dictated by the "PRINT=" variable in the SCHEMA.CFG configuration file.

Operation: Select **SCHPRINT** and the command lights. If a drawing has been preselected, the COMMAND text line displays:

SCHPRINT DRAWING

where **DRAWING** is the preselected drawing. Press **<Enter>** to invoke SCHPRINT with the selected drawing file (and options, if specified). SCHPRINT options can be specified after **DRAWING**, if desired. Refer to Chapter 4, *SCHPLOT/SCHPRINT*, for information on print options.

See Also: **DRAWINGS**; Chapter 4, *SCHPLOT/SCHPRINT*

Notes: A drawing must be selected before SCHPRINT can be invoked from the Manager. Refer to the **DRAWINGS** command description for information on selecting drawings from the Manager.

The "PRINT=" variable is unique to the SCHEMA.CFG file (as is the "GRAPH=" variable).

2.4.3 SCHEMA-PCB Commands

The following commands are installed to the Manager only if the SCHEMA-PCB Interface is installed. These commands access the PCB design, SUPERROUTER, and ECO features of SCHEMA.

SCHECO

Use: Forward annotate a SCHEMA-PCB design with SCHEMA III schematic changes or back annotate a SCHEMA III schematic with SCHEMA-PCB design changes.

Operation: Select **SCHECO** command lights. The COMMAND text line displays:

SCHECO DRAWING

where **DRAWING** is the preselected drawing (less extension). Press **<Enter>** to invoke SCHECO.

See Also: Chapter 8, *Utilities; SCHEMA PCB Layout User's Manual*, Chapter 9, *ECO*

SCHPCB

Use: Invoke SCHEMA-PCB.

Operation: Select **SCHPCB** and the command lights. The COMMAND text line displays:

SCHPCB

Press **<Enter>** to invoke SCHEMA-PCB.

See Also: *SCHEMA PCB Layout User's Manual*

SEDIT

Use: Text editor for creating or editing ASCII files.

Operation: Select **SEDIT** and the command lights. The COMMAND text line displays:

SEDIT FILENAME.ASC

where **FILENAME.ASC** is the name of the new file. Press **<Enter>** to load the editor screen.

See Also: **SEDIT**, *SCHEMA-PCB Layout User's Manual*, Appendix A, *ASCII Files and SEDIT*

SLIB

Use: SCHEMA-PCB User library and Standard library management utility.

Operation: Select **SLIB** and the command lights. The COMMAND text line displays:

SLIB

Press **<Enter>** to invoke the SLIB editor.

See Also: **SLIB**, *SCHEMA-PCB Layout User' sManual*, Chapter 13, *SCHEMA-PCB Part Library*

SREAD

Use: Convert a SCHEMA-SUPERROUTER .ROU file to a SCHEMA-PCB .JOB file..

Operation: Select **SREAD** and the command lights. The COMMAND text line displays:

SREAD <filename>

Press **<Enter>** to invoke SREAD.

See Also: *SCHEMA PCB Layout User's Manual*

S_ROUTER

Use: Invoke the SCHEMA-SUPERROUTER.

Operation: Select **S_ROUTER** and the command lights. The COMMAND text line displays:

S_ROUTER

Press **<Enter>** to invoke SCHEMA-SUPERROUTER.

See Also: *SCHEMA PCB Layout User's Manual*

SWRITE

Use: Convert a SCHEMA-PCB .JOB file to a SCHEMA-SUPERROUTER .ROU file.

Operation: Select **SWRITE** and the command lights. The COMMAND text line displays:

SWRITE <filename>

Press **<Enter>** to invoke SWRITE.

See Also: *SCHEMA PCB Layout User's Manual*

2.5 SCHEMA Manager Configuration

Section 2.5 is intended for those who wish to further customize the Manager or develop applications to compliment the Manager.

The SCHEMA.BAT, SCHEMA.MGR and S$.EXE files are used

Before changing the SCHEMA.BAT or the SCHEMA.MGR file, make a backup copy. This is to insure that you will be able to use the Manager if something should go wrong with the files that you are editing.

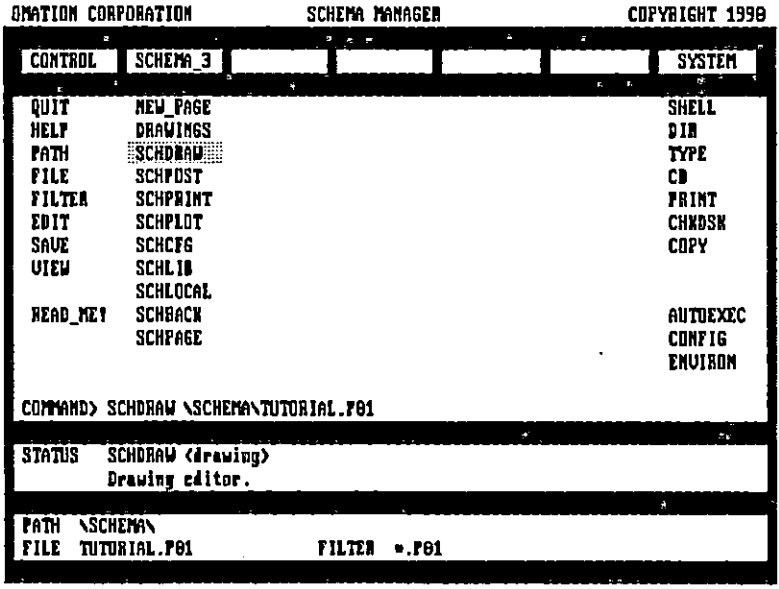


Figure 2-2. SCHEMA Manager.

2.5 SCREEN DESCRIPTION

Each screen consists of four parts as shown in figure 2-2. The title block contains "Omation Incorporated" and a copyright notice hardcoded. The title of the Manager ("Schema Manager" or a user selected title) appears centered in the title block area.

The header row consists of an area capable of holding seven headers for the seven columns. Each header is eight or fewer characters. The command columns are each capable of holding thirteen commands of eight or fewer characters.

The STATUS block shows the help messages associated with each command when pre-selected. These are replaced by error messages when an error has occurred. The status block also displays the current manager path, file and filter selected.

2.6 CUSTOMIZING THE MANAGER

The default Manager Configuration is the SCHEMA.MGR file. This file can be edited with any ASCII Editor or with the built-in Manager edit feature EDIT@D. If you should change the configuration file, be sure to make a backup copy before doing so.

The Header information begins with an open bracket ([ ) and ends with a closed bracket ( ]).

The first line of variables contains four numbers for the selected color of the screen background, borders, text and reverse background for the cursor, respectively. The next two numbers represent the horizontal and vertical sensitivity of the mouse horizontally and vertically. Each entry is followed by a comma except the last variable. For example, the line

should be entered as follows: **1,15,15,4,2,4.** The first four numbers are corresponding color numbers and the last two numbers are for the sensitivity of the mouse, X,Y respectively. Refer to section **2.6.1 COLORS.**

The **TITLE** line should be entered exactly as you would like the title of the Manager screen to appear. The default is SCHEMA MANAGER. The title is limited to 40 characters.

The header line contains the header names for the screen columns displayed across the top of the Manager screen. Each column header is followed by a comma except the last header name. If a column is to be skipped (not named), then just enter the delimiting comma with no entry. The header line should contain a total of six commas delimiting the seven headers.

The **PATH** line should contain the default path that is loaded when the Manager is invoked. For example, if your drawing files for SCHEMA III are contained in the directory called SCHEMA on drive C, then enter the default path as **C:\SCHEMA**. If your drawing files always reside in the current directory, **PATH** may be left blank.

Next, the **FILE** line is the default file or schematic drawing that will be loaded with the Manager. If you are working on a project with a specific drawing name, you can change the **FILE** default to be that drawing name. When the Manager is invoked, that drawing name will automatically appear as the default to be used with the different program applications.

The **FILTER** line is the filter default to be loaded. The line may contain the * and ? wildcard characters. For example, if you wanted to be able to load blank or created drawings automatically when the Manager is invoked, enter ***.P*** as the default. If you are more interested in just blank drawing pages, enter ***.P99**. If you are interested in all files, enter

..

The PATH, FILE and FILTER lines are optional and do not have to be used at all. However, a blank line for each line must appear in the configuration file. If PATH, FILE, and FILTER are assigned they will be automatically substituted for %1, %2, and %3, respectively, in the Manager Command Line.

The last line is then closed with the closed bracket (]).

The second format type is the **COMMAND** format. The first line begins the format with an open brace ({).

The second line is the X,Y coordinate that the specified command is located on the Manager screen followed by an optional color for the command to appear. For example, PATH is in the first column and the fourth command down that column, displayed in yellow. For this format enter: **1,4,14**. Refer to 2.6.1 COLORS for more information.

The **COMMAND TITLE** line is the name of the command itself. For example, the Schematic Drawing Editor program is titled SCHDRAW for the command name.

The **COMMAND PATH** is the path where the command resides. For example, SCHDRAW is contained in the SCHEMA directory on drive C. The entry for this path should be **C:\SCHEMA**.

The fifth line contains the **COMMAND** followed by its respective parameters. The parameters build the command for the particular requirements of that program. Refer to section **2.6.2 COMMAND LINE VARIABLES** and **2.6.3 EXECUTION PARAMETERS** for an explanation of each symbol.

The first parameter type is the Percent Variables which substitute the

current manager **PATH, FILE** and/or **FILTER** value. For the command to automatically includes the PATH, FILE, and FILTER values when the command is executed, include **%1, %2** and/or **%3** after the **COMMAND.** Remember that the **%2** has the same value as **#.#** described below. For example, the SCHDRAW program can load with the current **FILE** that is displayed in the Manager Status Block. If you enter: **SCHDRAW %2**, then the current file will be loaded when SCHDRAW is invoked.

The second type is the Pound Variables which substitute the name and/or extension of the current **FILE** parameter. If you want the command to automatically use any portion of the FILE parameter when executed, enter **#.#, #., .#** and/or **#.** For example, SCHCFG uses the current drawing name and changes the extension to .CFG. If you enter: **SCHCFG #.CFG**, the configuration file for the current drawing will be loaded when using SCHCFG.

The third type is the Execution Parameters which include Execution Method and Path Parameters. These variables instruct the command how to execute.

Execution Method parameters are System Commands, Executable Programs, Shell and Batch modes **(D, E, M, R, S).** Path Parameters are **F** and **P** which will execute at the **FILE** or **COMMAND** path. The default is the current directory if neither Path Parameters are specified.

For example, SCHCFG is a "Shell" program which executes at the COMMAND path with the current manager FILE and will return to the Manager screen when finished. For this type of execution enter: **SCHCFG #.CFG@PS.**

The fourth type is the Ampersand Variables which substitute the PATH, FILE and/or FILTER string long address in memory for executable

commands which need to update any of the parameters. For example, the DRAWINGS command shells and updates the current manager PATH and FILE values prior to returning control to the Manager. For executing DRAWINGS enter: **DRAWINGS %1%3@PS &1 &2.**

NOTE

The ampersand variables are provided for programmers and advanced users creating custom application programs which shell and require access to these Manager variables.

The sixth and seventh lines are **HELP** lines. These **HELP** message prompts are displayed when the command is highlighted. The first line usually contains the format and parameters that the command requires when invoked. *These do not have to be used at all.* If you do not care to have **HELP** lines, close the **COMMAND** format with a brace (}) on the line after the **COMMAND** and its parameters.

The format ends with the closed brace (}) on the last line.

2.7 CONFIGURATION FILE FORMAT

The .MGR configuration file is a user definable ASCII file which describes the appearance and operation of the Manager. Refer to section **2.6 Customizing The Manager Configuration** for more details. There are two concurrent formats used. The first is the **HEADER** information enclosed in square brackets as follows:

```
[
BACKGROUND,BORDERS,TEXT,REVERSE,XMOUSE,YMOUSE
TITLE
HEADER1,HEADER2,HEADER3,HEADER4,HEADER5,HEADER6,HEADER7
PATH
FILE (or DRAWING)
FILTER
]
```

The **BACKGROUND, BORDERS, TEXT** and **REVERSE** parameters set the colors for the menu. The **XMOUSE** and **YMOUSE** parameters set the vertical and horizontal mouse sensitivity, respectively. **TITLE** is the title of the manager. The headers (**HEADER1** through **HEADER7**) are listed in order with unused headers delimited. **PATH** is the default path that is loaded when the Manager is invoked. **FILE** (or DRAWING) is the default file (schematic drawing) that will be loaded with the Manager. **FILTER** is the default filter to be loaded (e.g., ***.***, ***.P99**, ***.P01**, ***.CFG**).

For example:

```
[
1,15,15,4,2,4
SCHEMA MANAGER
MANAGER,SCHEMII+,SCHPCB,INTEL,XILINX,,DOS
C:\SCHEMA\
TUTORIAL.P01
*.*
]
```

The second is the **COMMAND** format:

```
{
COL,ROW,FOREGROUND,BACKGROUND
<COMMAND TITLE>
<COMMAND PATH>
COMMAND %nparameter,#,#.EXT,NAME.# @<DEMRS><FPX><W> &n
HELP1
HELP2
}
```

Each structure starts with a left brace and ends with a right brace. The **COL** and **ROW** coordinates of the command title are second with an optional **FOREGROUND** color. The command title exactly as it will be displayed is next. The path where the command resides is next. The command and parameters follow. The execution parameters and ampersand command parameters appear at the end of the command line following the @. Command line parameters and execution parameters are described later. The next two lines are the help messages displayed when the command is pre-selected. Generally the first help message displays the command format with parameters listed.

For example:

```
{
1,3,14
PATH
C:\
PATH %1@D
PATH is a manager command.
The PATH parameter appears at lower left.
}
```

```
{
2,2,14
DRAWINGS
C:\INSTALL
DRAWINGS %1%3@PS &1 &2
HELP1
HELP2
}


{
2,3,14
SCHDRAW
C:\SCHEMA
SCHDRAW %1%2@RX
THIS IS THE FIRST HELP LINE
SO THAT MAKES THIS THE SECOND HELP LINE
}


{
2,7,14
SCHCFG
C:\SCHEMA
SCHCFG #.CFG@PS
SCHCFG <FILENAME>
SCHEMA CONFIGURATION UTILITY
}
```

```
{
7,1,14
COPY
C:\
COPY@RPW
}
```

2.7.1 COLORS

Color parameters will have no effect on the manager when used with monochrome monitors, monochrome intensities are default set in the software and cannot be altered. Users with color monitors can use the following guide in selecting colors. Textmode colors may vary slightly between monitors due to differences in hardware. In selecting colors for the manager, note that some colors may only be used for foreground (i.e., the **BORDERS, TEXT,** and **FOREGROUND** color parameters described above).

Available colors that can be used for the foreground and background:

BLACK	0
BLUE	1
GREEN	2
CYAN	3
RED	4
MAGENTA	5
BROWN	6
LIGHTGREY	7

Available colors that can ONLY be used for the foreground:

DARKGREY	8
LIGHTBLUE	9
LIGHTGREEN	10
LIGHTCYAN	11
LIGHTRED	12
LIGHTMAGENTA	13
YELLOW	14
WHITE	15

2.7.2 COMMAND LINE VARIABLES

The following parameters are used in actual command string to automatically substitute the manager path, file, and filter values at specific locations in the command string.

A. Percent variables:

%1 - substitute the current manager **PATH** string value.

%2 - substitute the current manager **FILE** string value.

%3 - substitute the current manager **FILTER** string value.

The **%2** parameter has the same value or meaning as ***#.#*** (see ***#.#*** below).

B. Pound variables:

#.# - substitute NAME.EXT of current **FILE** parameter.

#. - substitute NAME. of the current **FILE** parameter.

.# - substitute .EXT of the current **FILE** parameter.

- substitute NAME of the current **FILE** parameter.

C. Ampersand variables:

&1 - substitute the manager **PATH** string variable address for executable commands (**.EXE**) which shell and need to update the **PATH** parameter.

The address is provided as an 8 characterlong hex number string representing the 32-bit address where the PATH string variable is located.

&2 - substitute the **FILE** string long address.

&3 - substitute the **FILTER** string long address.

Refer to the DRAWINGS.EXE example in section X.XX, CONFIGURATION FILE FORMAT for an illustration of the usage and placement of ampersand variables in the command line. Placement after the @ parameters suppresses the display of & variables at the Manager command line but makes the variables available at execution time.

NOTE

The ampersand variables are provided for programmers and advanced users creating custom application programs which shell from the Manager and require access to these variables.

2.7.3 EXECUTION PARAMETERS

The following parameters appear at the end of each executable command line in the .MGR ASCII file. An @ will mark the end of the command line and the beginning of the Execution Parameters. The parameters dictate the type of command (manager dedicated, new manager, or executable), the paths which will be used in executing the command (command path, file path, or both), and the method by which the command is to be executed (batch, shell, or exec/no return).

A. Execution Method parameters:

D - dedicated manager command - **QUIT, PATH, FILE, FILTER, SAVE, EDIT, VIEW, HELP** when followed by @D are assumed to refer to the manager equivalents, otherwise they will be treated as executable commands outside the Manager.

E - EXEC the command leaving the manager, no return to the manager.

M - the command is actually a new manager configuration to be loaded.

R - execute via batch mode and RETURN, manager does not remain resident but returns after command execution.

S - execute by SHELLing the command in the remaining memory, the manager stays resident and returns after execution.

default - if none of the above follow the @, the commands are treated as executable commands outside the Manager and executed via the batch

mode (@R).

B. Path parameters:

F - execution will occur in the FILES (current manager PATH variable) directory.

P - execution will occur in the PROGRAM directory (<COMMAND PATH> from .mgr configuration file).

X - execution will occur in the current directory.

C. Miscellaneous parameters:

W - WAIT before returning to manager in batch mode execution only (see R above) to view the result of a command execution. This parameter executes the DOS PAUSE command prior to recalling the Manager.

D. Usage:

For commands which do not load a new manager (@M) or perform specific (dedicated) manager related tasks (@D), parameters may be assigned at the end of the command line preceeded by an asterisk by selecting one item from the top row of the matrix (execution method parameter) and one item from the left column of the matrix (path parameter). The default execution method is @RX and is marked in the matrix with an *. Note that W can be selected in conjunction with R if so desired.

	R(W)	S	E
P			
F			
X	*		

NOTES

SYSTEM COMMANDS. SYSTEM commands (files with extension .COM) and BATCH files (files with extension .BAT) must be called in a batch fashion from the manager (i.e., @RP, @RF, or @RX). The W option may be added as well if it is necessary to view the result of execution prior to returning to the manager.

EXECUTABLE PROGRAMS. Compiled executable programs (files with extension .EXE) can be called in a batch mode fashion (as above) or may be "shelled" leaving the manager code resident.

BATCH VS. SHELL. Executing in a batch mode frees all of memory to the command or program being executed. Shelling a command causes the manager to remain resident and the command to be executed in the remaining base memory. The advantage to shelling is speed in returning to the manager since it does not need to be reloaded from disk.

DOS only recognizes the most recent batch file invoked. Applications calling batch files will not return to the Manager unless the Manager is re-called from their batch files.

<Left Blank>

Chapter 3. SCHDRAW

3.1 INTRODUCTION

This chapter describes the SCHDRAW program and its commands.

SCHDRAW is the major component of SCHEMA III, providing the graphics interface for drawing schematics and creating/editing objects. SCHDRAW is the only program included with SCHEMA III that provides for graphical drawing and object creation. SCHDRAW contains two unique graphics editors: *Schematic* and *Object.*

The Schematic Editor is used to draw schematics (and other drawings) while the Object Editor is used to create/edit objects used in the Schematic Editor.

The Schematic and Object Editors are highly-interactive programs controlled by the keyboard or a mouse. Various primary and secondary menu commands are selected with the keyboard or mouse and then executed. All line, object, and text placement is done in real-time using the keyboard or a mouse.

The Schematic Editor uses drawing files as both its input and output. The input drawing file is a blank drawing page or existing drawing. From a blank or existing drawing page, lines, text, and objects are added to form the desired drawing file which is then saved to disk. The SCHPRINT, SCHPLOT, and SCHPOST programs use completed drawing files for input data. Only the SCHDRAW Schematic Editor can create or update a drawing file.

The Object Editor allows existing objects to be edited and new objects to be created. Most of the editing features used to create schematics in the Schematic Editor are available in the Object Editor to create objects.

3.2 INVOKING SCHDRAW

Like most SCHEMA III programs, SCHDRAW can be invoked from the Manager or directly from the DOS prompt. Chapter 2, *SCHEMA Program Manager*, contains detailed information on the Manager and the procedure to invoke SCHDRAW. Whether you use the Manager or DOS, the SCHDRAW command line is:

SCHDRAW DRAWING

where **DRAWING** is the optional name of the schematic drawing to load. **DRAWING** must specify the complete path and file name of the desired drawing. If no extension is included, SCHEMA III loads the drawing with the .P01 file extension. After the SCHDRAW command line is entered, press **<Enter>** to invoke SCHDRAW with the specified drawing. Some sample SCHDRAW command lines are:

SCHDRAW B:POWER.P03 <Enter> (Invokes SCHDRAW with drawing POWER.P03 on Drive B.

SCHDRAW \MEMORY\POWER.P03 <Enter> (Invokes SCHDRAW with drawing POWER.P03 from directory \MEMORY.

SCHDRAW POWER <Enter> (Invokes SCHDRAW with drawing POWER.P01 from the current directory)

After SCHDRAW is initialized, the following message is displayed:

SCHEMA DRAW Version X.XX Copyright Omation, Inc 1988

(where: **X.XX** is the current version number.).

This message only appears briefly and then SCHDRAW finishes loading.

3.2.1 SCHDRAW Screen Format

After SCHDRAW loads, the screen is initialized with the Schematic Editor. Figure 3-1 shows a typical Schematic Editor screen. Up to five windows and the cursor are displayed. The cursor is represented by a small arrowhead and is controlled by the keyboard or a mouse.

The Command Window displays associated editor commands within Command Boxes. The commands in the upper-half of the Command Window are called Primary Commands. Some Primary Commands have additional, subordinate commands called Secondary Commands, which appear in the lower-half of the Command Window when the associated Primary Command is activated. Lighted Command Boxes indicate selected, active commands. Figure 3-1 shows the **DRAW** Primary Command and its associated **OBJECT** Secondary Command active.

The Object Editor is invoked by selecting the Schematic Editor **FILE** Primary Command, followed by the **ED OBJ** Secondary Command. The Object Editor is used to create and edit objects used in the Schematic Editor. A typical Object Editor screen is shown in figure 3-2. Paragraph 3.6 describes each Schematic and Object Editor command.

The Drawing Window displays a portion of the current drawing or object. All drawing activities, i.e., adding lines, text, objects, etc. occur in the Drawing Window. When the cursor is moved to any edge of the screen, the Drawing Window pans in the appropriate direction.

The Text Window appears only when keyboard entry is required or a special prompt is displayed. Text is displayed in the Text Window as it is entered from the keyboard. Text Window editing keys and their functions are covered in Appendix C, *Text Editing Keyboard Usage.*

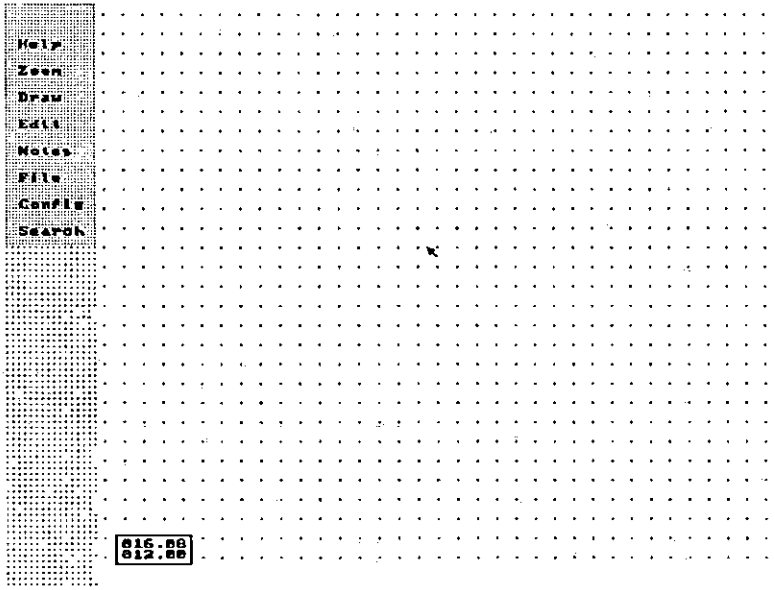


Figure 3-1. The SCHDRAW Schematic Editor

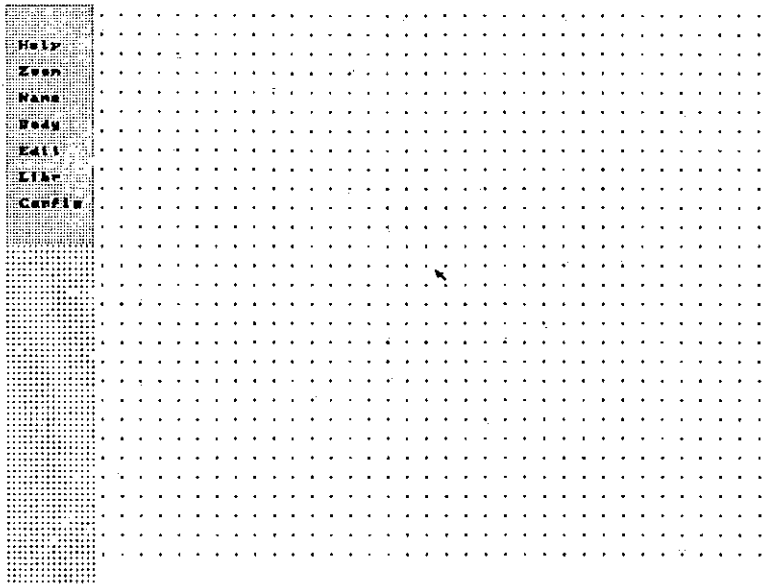


Figure 3-2. The SCHDRAW Object Editor

The Error Window displays error messages and appears only when an error occurs. If the Error Window appears, press any key to acknowledge the error and remove the Error Window. Refer to Appendix D for a summary of SCHEMA III error messages.

The Cursor Coordinate Window (not shown) displays the current cursor position relative to a 16-pixel visual dotted grid.

3.2.2 Moving the Cursor

The cursor is controlled with the keyboard or a mouse. Mouse operation is very simple - moving the mouse moves the cursor.

To move the cursor with the keyboard, use the **<Arrow>** keys. If the Cursor Grid (**CONFIG-C GRID**) is active, each **<Arrow>** keypress advances the cursor on or half-way between the Visual Grid (**V GRID**) dots. If **C GRID** is off, each keypress advances the cursor 1 pixel (1/16 Visual Grid dot). The cursor can be jumped to a left or right Drawing Window border by simultaneously pressing the **<Ctrl><Arrow>** keys. For example, when **<Ctrl><Right Arrow>** is pressed, the cursor jumps to the right edge of the Drawing Window and the drawing pans right. Pressing **<Home>** moves the cursor to the center of the Drawing Window. The **<PgDn>** key jumps the cursor to the bottom edge of the Drawing Window and **<PgUp>** jumps the cursor to the top edge. Table 3-1 contains a summary of SCHDRAW keyboard operation.

3.2.3 Selecting Commands

SCHDRAW commands are selected using the cursor (controlled by the keyboard or a mouse), the middle mouse button (three-button mouse users only), or the keyboard.

3.2.3.1 Using the Cursor to Select Commands

To select commands with the cursor, use the keyboard or a mouse to position the cursor over the desired command box. Next, either press **<Ins>** twice or click the left mouse button to select the command.

Table 3-1. SCHDRAW Keyboard Usage Summary

KEY	*FUNCTION*
<Up Arrow>	Moves cursor up
<Down Arrow>	Moves cursor down
<Left Arrow>	Moves cursor left

<Right Arrow>	Moves cursor right
<Ctrl><Left Arrow>	Jumps cursor to left edge of Drawing Window
<Ctrl><Right Arrow>	Jumps cursor to right edge of Drawing Window
<PgDn>	Jumps cursor to the bottom edge of the Drawing Window
<PgUp>	Jumps cursor to the top edge of Drawing Window
<Home>	Jumps cursor to center of Drawing Window
<F1>	Next keypress is first letter of desired Primary Command
<F2>	Next keypress is first letter of Secondary Command
<F3>	Redisplays last text string entered in Text Window
<F4>	Turns **CONFIG-DIAG** command on or off
<F5>	Turns Cursor Coordinate Window on or off
<F7>	Zooms In one level
<F8>	Zooms Out one level
<F9>	Repeats previous command
<F10>	Turns Cursor Grid (**C GRID**) on or off
<Ins>	Simulates pressing the left mouse button to select a command or do an editing function
<Esc>	Activates **UNDO** command

SCHDRAW uses the left mouse button for command selection. *Unless otherwise stated, all references to the mouse button in this manual refer to the left mouse button.*

3.2.3.2 The Middle Mouse Button

If you have a three-button mouse, the center (middle) mouse button can also be used for command selection. Clicking the middle button freezes the cursor's position (cursor disappears) in the Drawing Window and activates the Command Window Indicator. The Command Window Indicator erases the right and left sides of the currently selected command box. As the mouse is moved, the Command Window Indicator moves up and down the Command Window. When the desired command box is selected, click the left mouse button. Next, click the center mouse button to reactivate the cursor in the Drawing Window.

3.2.3.3 Using the Keyboard to Select Commands

To select a SCHDRAW Primary Command with the keyboard, press the <F1> Function key. The <F1> key informs SCHDRAW that a Primary Command is about to be selected. Next, press the Capital letter of the desired Primary Command Box. To select a Secondary Command, simply press the Captial letter of the desired Secondary Command Box *after the associated Primary Command has been selected.* If the desired Primary Command has been selected to perform a Secondary Command, the Primary Command does not need be reselected to perform another secondary command. If the Text Window is active, all keypresses are displayed in the Text Window. To override the Text Window, press the <F2> Function key followed by the Capital letter of the desired Secondary Command.

3.2.3.4 Auto Backup

In the event of a power failure or software hang you have a backup. SCHEMA backups up your schematic every five minutes. The file updated is called, BACKUP.P01. SCHEMA will use the OMATION environment variable to find the path to where to send the file. The interval between backups can be adjusted by using SCHCFG to change

the SCHEMA.CFG *ONLY*. The variable is located in the Drawing Parameters Configuration. The numeric values are integer and in minutes. The default is five, but you can raise or lower this value. A value of **0** will disable this feature.

3.2.4 Editing With a Mouse/Keyboard

Most SCHDRAW editing functions use the left mouse button to start/stop an editing procedure. Pressing the left button activates a function, while releasing the button stops the function. The keyboard **<Ins>** key operates in a similar manner. Pressing **<Ins>** once is the same as pressing the left mouse button while pressing **<Ins>** again is the same as releasing the left mouse button.

If you try to select a command while the left mouse button is pressed (or after **<Ins>** is pressed once), a long beep is generated. Repress **<Ins>** to simulate releasing the mouse button and reselect the desired command.

3.2.5 UNDO

There are two ways to activate an UNDO sequence. First, UNDO can be executed by pressing the **<ESC>** key, or by clicking the right mouse button. SCHDRAW will undo an Undo and will toggle back and forth between them.

3.3 USING THE SCHDRAW SCHEMATIC EDITOR

The following paragraphs discuss how to use the SCHDRAW Schematic Editor to draw and print schematics. The term 'schematic' refers to a set of pages that collectively comprise a single drawing.

3.3.1 Blank Pages

SCHEMA III is shipped with several blank drawing pages which appear as DOS files. These blank pages are used to establish the desired virtual drawing page size when a new drawing is started. The following blank pages are supplied with SCHEMA III:

PPPAG1.P99 - Virtual C-size Totally Blank Page, Portrait
PPAGE1.P99 - Virtual C-size Totally Blank Page, Landscape
BPAGE1.P99 - Virtual C-size Bordered Blank Page, Landscape
PPPAGE2.P99 - D-size Standard Blank Page, Portrait
PPAGE2.P99 - Virtual D-size Totally Blank Page, Landscape
BPAGE2.P99 - Virtual D-size Bordered Blank Page, Landscape
PPPAGE3.P99 - E-size Standard Blank Page, Portrait
PPAGE3.P99 - Virtual E-size Totally Blank Page, Landscape (Requires 640K RAM)
BPAGE3.P99 - Virtual E-size Bordered Blank Page, Landscape (Requires 640K RAM).

These pages are designed for printers at 60 dpi:

PPAGEA.P99 - A-size Totally Blank Page
PPAGEB.P99 - B-size Totally Blank Page
PPAGEC.P99 - C-size Totally Blank Page
PPAGED.P99 - D-size Totally Blank Page
PPAGEE.P99 - E-size Totally Blank Page + 1Meg EMS
BPAGEA.P99 - A-size Totally Bordered Page
BPAGEB.P99 - B-size Totally Bordered Page
BPAGEC.P99 - C-size Totally Bordered Page
BPAGED.P99 - D-size Totally Bordered Page
BPAGEE.P99 - E-size Totally Bordered Page + 1Meg EMS
PPPAGEA.P99 - A-size Totally Blank Page, Portrait
PPPAGEB.P99 - B-size Totally Blank Page, Portrait
PPPAGEC.P99 - C-size Totally Blank Page, Portrait

PPPAGED.P99 - D-size Totally Blank Page, Portrait
PPPAGEE.P99 - E-size Totally Blank Page, Portrait + 1Meg EMS

Non-standard page sizes, including European-standard engineering pages can be created using the SCHPAGE utility described in Chapter 8, *Utilities*.

The actual, final output drawing size is determined by the type of output device used (printer or plotter) and the print mode selected (A, C or D). Printing is discussed later. Notice that bordered pages are named similar to blank pages, except they begin with B. For example, BPAGE2.P99 is a D-sized, landscape bordered page. SCHEMA III's bordered pages can be edited to meet your company's requirements. If you do not like the border arrangement, use the **EDIT-ERASE** and **NOTES-LINE/TEXT** commands to redraw the border, as required. See **Customizing Pages For a Particular Printer** and **Customizing Pages For a Particular Plotter**, *Chapter 4 - SCHPLOT/SCHPRINT*

If you wish to permanently alter a blank page so that each time you assign a project name to it, the new page will have your changes, then call the blank page with SCHDRAW like any other file. Alter the page and execute **FILE-SAVE**. Upon exiting this blank page will have your alterations imbedded into it.

The SCHEMA Program Manager allows for quick blank page selection. After the desired blank page is selected, the Manager makes a copy of the page and automatically renames it as the desired drawing name.

If the Manager is not used, invoke SCHDRAW with the desired blank page, create your drawing, and then save the page as the desired drawing name. If another page is required, simply load a new blank page, draw the drawing, and save the page.

The first page of a schematic drawing is always saved as DRAWING.P01 (where .P01 identifies page 1), DRAWING.P02 is page 2 of the drawing, and so on.

3.3.2 The Drawing Title Block

Most schematics contain a drawing title block. The drawing title block contains information such as the drawing name, drawing number, revision number, date, maximum page number, etc. Additional information can also be displayed within the title block, as required. As shipped, the title blocks are located in the lower, right corner of the blank, bordered pages. If you do not like the title blocks or their locations, erase and edit as required.

With SCHDRAW, you can create any type of title block using the NOTES-LINE command. Figure 3-3 shows a typical SCHEMA III drawing title block.

Every time a new schematic is started, you can create a configuration file to contain the drawing's title block information. The Manager (discussed in the tutorial in Chapter 1 and Chapter 2) lets you create the drawing's configuration file quickly. The Manager uses the SCHCFG program, discussed in Chapter 5, to create the drawing configuration file and pre-assign drawing title block information to your title block.

The SCHCFG **F1 - F8** title block variables let you define your title block contents once without having to enter the same information every time a new blank page is loaded. Then, when your blank drawing pages are loaded, the predefined title block information is automatically placed in the desired locations within the drawing title block. This greatly simplifies title block set-up and maintains consistency between schematic pages. The need to manually enter title block data for each separate schematic page is eliminated by this labor-saving convenience. Chapter 5, *SCHCFG*, outlines the procedure to preassign drawing title

block information. The SCHCFG **F1 - F8** title block variables are also printed along the top of several SCHPOST outputs to associate the post-processing reports/outputs with the applicable drawing.

SCHDRAW allows you to place pointers to the variables and have them automatically updated when the variables are changed in the CONFIG files. Use the **NOTES-TEXT** command. For example, if you wanted to use one of the currently unused block variable, **F8**. On the **NOTES-TEXT** command window type:

<F8>..... <Enter>

Type the brackets and the letter 'F' with the number '8'. Do NOT type the function key. The extra dots are to handle variables greater than eight characters. There are three other Block Variables that are automatic and are not changed through SCHCFG. These are :

<FT> - Time and Date Stamp
<FF> - PATH and FILE name
<FP> - Schematic page number (**2** of 6)

These variables are updated when printing, plotting, or loading a drawing is done.

3.3.3 Object Libraries

All SCHEMA III objects are contained in object libraries. Up to nine different object libraries can be assigned to a schematic. Each object library can contain up to 4,640 total objects. Thus, up to 41,760 objects could be associated with a single schematic drawing set. The SCHCFG program, described in Chapter 5, is used to assign object libraries to your schematics.

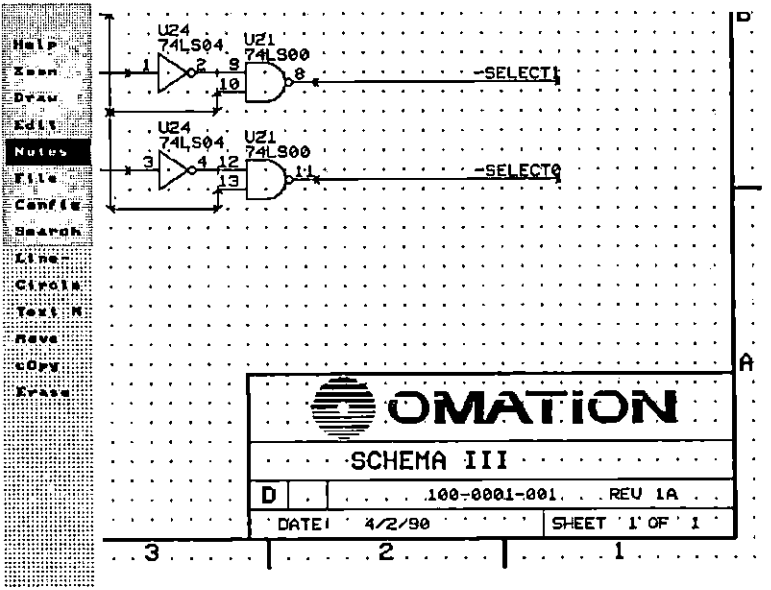


Figure 3-3. Typical Drawing Title Block

The GRAPH.LBR object library contains general-purpose objects such as arrows, flow chart symbols, assembly drawing outlines, etc.

IEEE.LBR contains all the IEEE/IEC symbols for most components.

Appendix A, *Installing SCHEMA III*, contains the procedure to install libraries.

Chapter 5, *SCHCFG*, outlines the procedure to assign object libraries to your schematics. Chapter 6, *SCHLIB*, describes how to create your own libraries. Appendix F, *Object Libraries*, further describes the libraries included with SCHEMA III and lists the contents of each library.

3.3.4 Basic Schematic Drawing Techniques

The process of creating general-purpose drawings such as block diagrams, assembly drawings, flow charts, etc. with SCHDRAW's Schematic Editor is fairly simple. Lines, text, and objects are placed in the Drawing Window with the keyboard or a mouse. There are no special rules or conventions to follow. However, when drawing *schematics*, you must adhere to a few rules. These rules ensure that the SCHPOST post-processing programs derive accurate component and connectivity information from the applicable schematic. The following paragraphs outline the proper procedures to draw schematics to ensure accurate SCHPOST post-processing.

3.3.4.1 Drawing Aids

There are several features available to the SCHEMA III user to aid the schematic drawing process: **HELP**, *Wire Headers* (**CONFIG-WI HDR**), the *Cursor Coordinate Window*, and both a *Visual Grid* (**CONFIG-V GRID**) and invisible *Cursor Grid* (**CONFIG-C GRID**). Together, these features greatly aid schematic creation.

The **HELP** command displays a listing of all the keyboard keys (and their functions) recognized by SCHDRAW. Use **HELP** if you do not know which keys can be used with SCHDRAW. **HELP** can also display a screen of information for each command by selecting the command boxes.

The **CONFIG-WI HDR** mode is extremely helpful since it clearly displays all signal wire end points, object connection points, and label assignment points (the exact location labels are placed). Figure 3-4 shows a Schematic Editor display with **WI HDR** enabled. Figure 3-5 shows the same display with **WI HDR** disabled. Keep **WI HDR** turned on when drawing schematics.

The Cursor Coordinate Window and Visual Grid (CONFIG-V GRID) are related. The Cursor Coordinate Window displays the current cursor position relative to a dotted, visual grid displayed by V GRID over the Drawing Window. Each visual grid dot is spaced at a 16-pixel interval. The Cursor Coordinate Window displays cursor position with respect to an X/Y coordinate system, beginning with 0,0 in the upper left corner of the Drawing Window. The SCHPOST post-processing error reports refer to the Cursor Coordinate Window coordinate to provide a convenient way to locate trouble spots.

The invisible Cursor Grid (C GRID) snaps cursor movement to an invisible eight-pixel grid. When the C GRID is on, cursor movement is restricted to on or half-way between the V GRID dots. *Always keep C GRID on when drawing schematics.* The SCHPOST post-processing programs cannot recognize connections made without the C GRID.

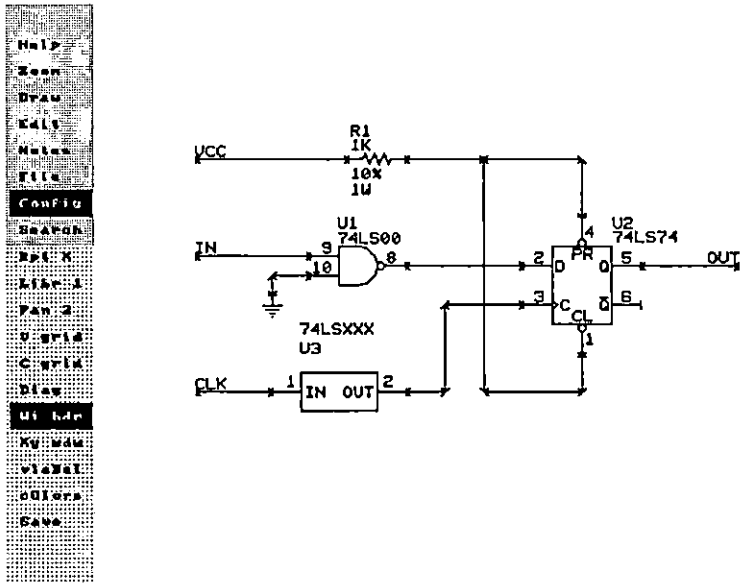


Figure 3-4. CONFIG-WI HDR Enabled

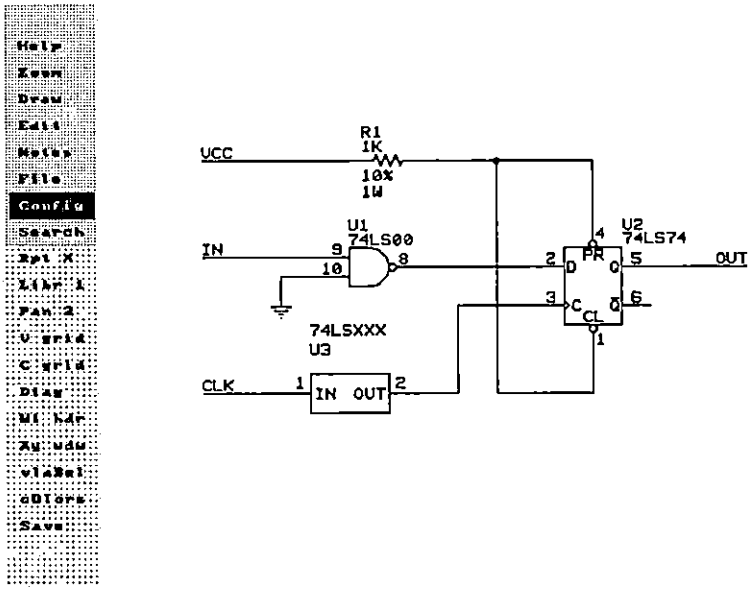


Figure 3-5. **CONFIG-WI HDR** Disabled

3.3.4.2 Using Components

Components are retrieved from SCHEMA III's object libraries using the **DRAW-OBJECT** command. When **DRAW-OBJECT** is selected, the Text Window is displayed so you can enter the **DRAW-OBJECT** *Parameter String.*

The **DRAW-OBJECT** Parameter String tells the Schematic Editor which component you want, and lets you assign a reference designator and up to four other object parameter fields to the component. The component name (such as 74LS00, LM555, 8048, etc.) is always entered first, followed by a comma and the desired reference designator.

For example:

74LS00,U2 <Enter>

tells the Schematic Editor you want the A-gate section of a 74LS00 component with an assigned reference designator of U2.

BROWSE

Another way to retrieve components is to use the parts browsing feature by selecting the **DRAW-OBJECT** command. Enter the following for example:

1=*

This will display a list of all the parts contained in the library assigned to LIBR1 of the CONFIG file. LIBR1 is default, but other LIBR numbers can be used. They will be *displayed on the screen in alphanumeric order.* The first part will be highlighted. To select a part, move the cursor with

the mouse or arrow keys. **<Home>** moves the cursor to the first name on the screen. **<PgUp>** will move the cursor to the previous screen and **<PgDn>** will advance to the next screen.

Press the left mouse or **<Ins>** key to retrieve the highlighted part. After a slight pause, the part will appear. If the object does not fit on one screen, use only the arrow keys to scroll the object. Press the right mouse or **<Esc>** key to return to the Directory Screen. Then, press the right mouse or **<Esc>** to return to the original drawing screen with the original library displayed in the Text Window.

If the object is one that you would like to use in your drawing, press the left mouse or **<Enter>**. There will be a slight pause and the drawing will appear with the object name in the Text Window. Press **<Enter>** and place the object with either the left mouse button or the **<Ins>** key. The parameters can be defined at this point just as if you used the **DRAW-OBJECT** command and entered the object name in the Text Window.

3.3.4.2.1 How to Select Different Component Gate Sections, Rotated Orientations, and DeMorgan Equivalents

Individual gates of multi-gate components, rotated components, and DeMorgan-equivalent components are easily retrieved from SCHEMA III's object libraries using either the **DRAW-GATE**, **DRAW-ROTATE**, or **DRAW-OBJECT** command.

The **DRAW-GATE** and **DRAW-ROTATE** commands automatically increment through all the available versions of a part. They are convenience features that automate the editing of the **DRAW-OBJECT** command line as will be described next.

To select a given gate section of a multi-gate device, simply enter the component's name followed by a backslash character (\) and alphanumeric sequence indicating orientation and gate section. For example, from **DRAW-OBJECT:**

74LS00\C,U23 <Enter>

selects the standard C-gate section of a 74LS00 and assigns a reference designator of U23 to the device.

Rotated representations of all *discrete* devices included with the SCHEMA III standard object library are easily selected. For example:

DRES,R1 <Enter>

selects a vertical resistor, while:

DRES\2,R1 <Enter>

selects a horizontally-oriented resistor.

DeMorgan equivalents are selected as follows:

74LS00\2C,U23 <Enter>

where **2** selects the DeMorgan-equivalent and **C** selects the C-gate section of a 74LS00.

Rotated versions of all gated parts are supplied with the standard libraries. You can select orientations 3-8 for all gates included in the standard SCHEMA III libraries. As previously discussed, the standard libraries supply orientations 1 and 2, where orientation 1 is the normal representation (oriented left-to-right) and orientation 2 is the

DeMorgan representation (oriented left-to-right). Rotated versions of the parts supply the following additional orientations:

3 - Normal, oriented bottom-to-top
4 - DeMorgan, oriented bottom-to-top
5 - Normal, oriented right-to-left
6 - DeMorgan, oriented right-to-left
7 - Normal, oriented top-to-bottom
8 - DeMorgan, oriented top-to-bottom

For example, to call up the right-to-left orientation, B-section of a 74LS00, select the SCHDRAW **DRAW-OBJECT** command and enter:

74LS00\5b<Enter>

while the top-to-bottom DeMorgan, C-section is retrieved by entering:

74LS00\8C <Enter>

3.3.4.2.2 Reference Designators

All component-type schematic objects are assigned reference designators. If you do not include a reference designator after the **DRAW-OBJECT** Parameter String component name/gate section entry, the Schematic Editor automatically assigns one to the component. Automatically-assigned reference designators can easily be changed using the **EDIT-IMAGE** command. When **EDIT-IMAGE** is used, the applicable object's **DRAW-OBJECT** Parameter String is displayed and can be edited. Also if you wish to change a group of parts use the **SEARCH** command. It will consecutively bring up each part in a group and give the Parameter String to edit.

When assigning component reference designators, always use alphanumeric characters. *Do not use spaces or dashes as part of the reference designator.*

Automatic reference designators consist of five characters: a period, three alphanumeric characters, and two numeric characters. The three alphanumeric characters begin with AAA (assigned to the first object with a blank reference designator field) and are incremented with each succeeding object. The two numeric characters represent the applicable schematic page number and are updated to reflect the actual page number when the schematic page is saved with the desired page number extension.

When an existing schematic is loaded and updated, any new automatic reference designators are assigned beginning with the next available alphanumeric numbering sequence. For example, if the last automatically-assigned reference designator was .AAC01, the next automatic reference designator is .AAD01, followed by .AAE01, etc.

Thus, all automatically-assigned reference designators are unique across multiple pages and after multiple edits. If pages are merged together using the **EDIT-REGION** command, all reference designators are automatically reassigned to ensure no reference designators are repeated in the drawing.

3.3.4.2.3 Assigning Other Parameters to Components

In addition to the component name and reference designator parameters, up to four other parameters can be included in the **DRAW-OBJECT** Parameter String. For digital components, the name and reference designator parameters usually carry the desired information. However, for discrete components (resistors, diodes, transistors, etc.), part values, tolerances, power ratings, and other component-related parameters are normally shown with the component on the schematic.

The four remaining parameters are used to display this information. For example, the **DRAW-OBJECT** Parameter String for a resistor can be expressed as:

DRES,R1,10K,1%,1/4W,XXX-XXXX-XXX <Enter>

This retrieves a **10K-Ohm, 1%, 1/4-watt** resistor with an assigned reference designator of **R1**. The last parameter entry, **XXX-XXXX-XXX**, is the resistor's part number, as determined by your company's part numbering system. This part number is never displayed with the resistor object but is saved for SCHPOST's post-processing programs.

How can you tell what parameters have been preassigned by Omation to discrete components and the order they should be entered in the **DRAW-OBJECT** Parameter String? Use the **EDIT-PM OBJ** command to display this information. Simply select **EDIT-PM OBJ** and position the cursor over the desired component. Next, either press **<Ins>** (keyboard users) or the mouse button. The Text Window appears with the component's parameter types.

For example, when **EDIT-PM OBJ** is used on a resistor, the Text Window displays:

NAME,REFD,VALUE,%%%%,WATT,

This identifies the **DRAW-OBJECT** Parameter String entry types for a resistor as the component's name, followed by the reference designator, part value, tolerance as a percentage, and wattage rating.

The last parameter is assumed to be your company's part number for the applicable discrete component. Note that **EDIT-PM OBJ** displays parameter types in the order they should be entered in the **DRAW-OBJECT** Parameter String. The physical locations of the various parameters, with respect to components, are determined in the Object

Editor, discussed later. If you do not like the physical positioning of the various component parameters, you can easily change them using the Object Editor.

3.3.4.3 Making Connections for SCHPOST

There are three ways to make or show connections in the Schematic Editor: signal **WIREs**, **LABELs**, and **BUSES**. Each connection has its own set of rules to ensure accurate SCHPOST post-processing. The following paragraphs describe these connections and the procedures required to make them recognizable by SCHPOST.

3.3.4.3.1 Signal Connections

Signal connections are the only type of electrical connection made on SCHEMA III schematics recognized by the SCHPOST post-processing programs. Signal connections connect individual component pin stub (header) end points with signal wires or signal wires with signal wires. Signal wires are drawn using the **DRAW-WIRE** command in the signal (thin) mode.

When connecting signal wires to component pins, do not overlap the signal wire with the component pin or draw the signal wire too short. Signal wires must terminate *exactly at the end of the component pin.* When bending or extending signal wires, begin the new signal wire section exactly where the original signal wire ends with no overlapping or gaps.

When connecting multiple components to form a single net, junctions are used. *SCHPOST recognizes only connections of at least three.* Use the **DRAW-DOT** command to add connection dots. Dots are added after 'T' or 4-way junctions are drawn, i.e., dots are added last. Figure 3-6

illustrates correct signal connection junctions. Figure 3-7 illustrates incorrect junctions.

The signal wire connection process is greatly simplified using the Cursor Grid (CONFIG-C GRID) and Wire Headers (CONFIG-WI HDR). When C GRID is used with WI HDR, it is difficult *not* to make proper signal wire connections.

3.3.4.3.2 Label Connections

A common label can be added to two or more individual signal wires to identify that the wires are connected. These wires can be on different pages of a schematic or on the same page. Labels are created using the DRAW-LABEL command and can be edited using the EDIT-IMAGE command. Part of the LABEL command involves selecting the proper label orientation (orientation refers to the relationship between the label and the signal wire to which it is assigned).

There are eight label/wire orientations:

a. A horizontal label to the left and above the wire
b. A horizontal label to the right and above the wire
c. A horizontal label to the left and centered on the wire
d. A horizontal label to the right and centered on the wire
e. A vertical label above and to the right of the wire
f. A vertical label below and to the right of the wire
g. A vertical label centered above the wire
h. A vertical label centered below the wire.

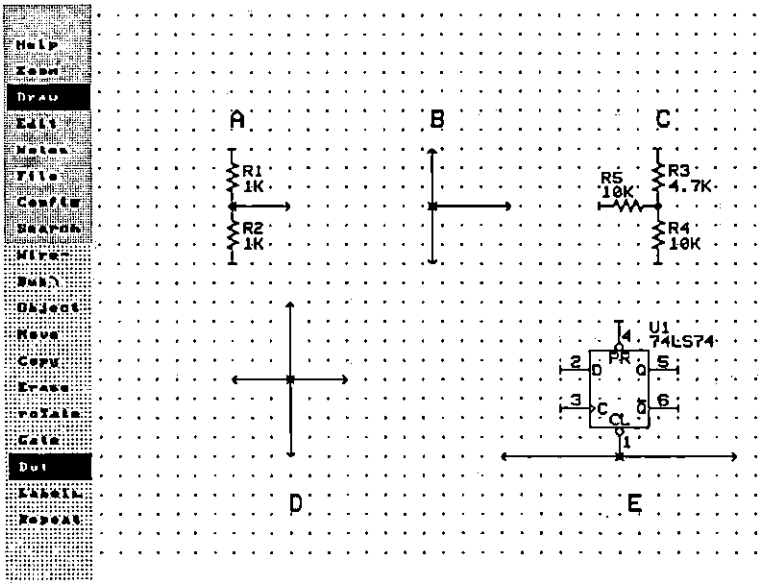


Figure 3-6. Correct Signal Connection Junctions

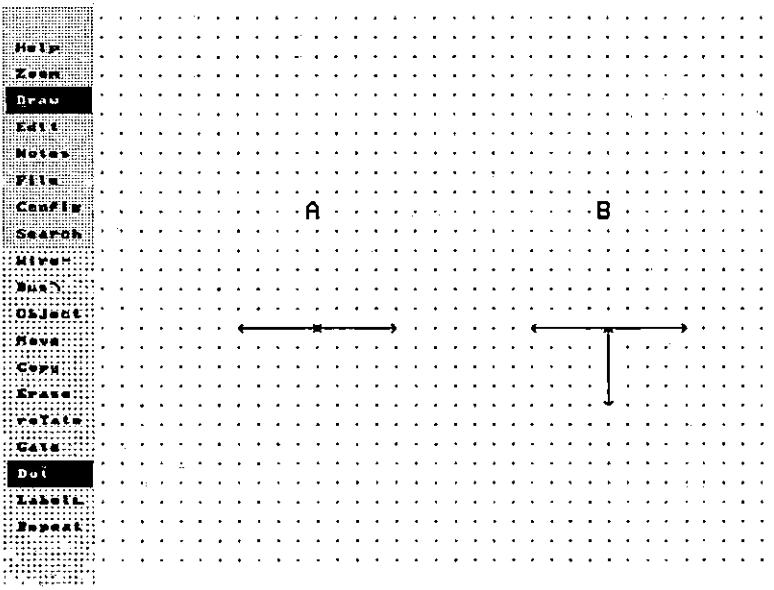


Figure 3-7. Incorrect Signal Connection Junctions

See the **DRAW-LABEL** command description in paragraph 3.5 for the methods used to select label orientation. Figure 3-8 shows these eight label orientations. This illustration shows **WI HDR** active. Thus, label assignment points and signal wire end point relationships are clearly visible. After the label orientation is selected and the label is being placed, a small line (when **WI HDR** is enabled) or dot is displayed with the label. This dot/line must be aligned with the exact end point of the signal wire to which the label connects. Again, **C GRID** and **WI HDR** greatly simplify this process.

It is sometimes desirable to label a signal wire somewhere along the length of the wire *other than the beginning or end point.* However, SCHPOST still requires the signal wire to start/stop at the label connection point. This is easily accomplished using the **DRAW-DOT** command. By placing a **DOT** on a label assignment point and then **UNDO**ing the **DOT** immediately after it is placed, the signal wire automatically splits at the **DOT** location, i.e., adding a **DOT** to a signal wire always creates a signal wire beginning/end point junction at the **DOT** location. Figure 3-8 shows one of these labels (**LABEL9**).

Power (normally +5, Vcc, Vss, etc.) is also assigned as a label. The only rule associated with power labels is to ensure your power label assignments match existing component power nets. For example, Omation has predefined the Vcc net for digital components as +5.

Thus, if you want to tie a component to +5, be sure to use the label +5, not +5V, +5.0, +5.0V, etc. Label swapping is discussed in Chapter 7.

Also, several different types of ground symbols are included in the SCHEMA III object libraries. These ground symbols actually assign a label at the connection point. They are:

GND - Digital Ground
GNDA - Analog Ground
GNDC - Chassis Ground

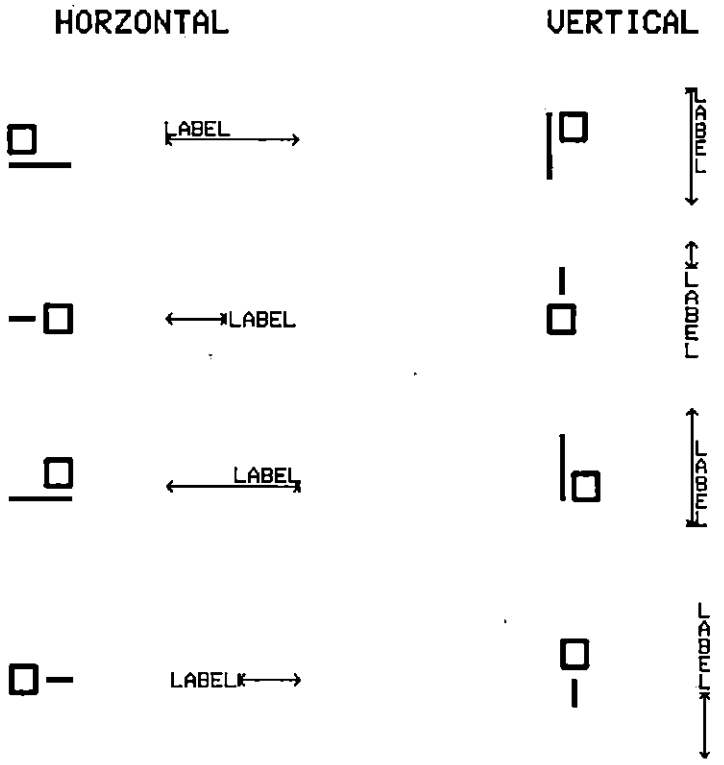


Figure 3-8. **DRAW-LABEL** Orientations

The name of the ground object is similar to the label name, i.e., object GND applies label GND, GNDA applies label AGND, and GNDC applies label CGND.

3.3.4.3.3 Bus Connections

Bus connections are used when multiple signal wires must be routed to a common location. Bus connections improve schematic legibility by reducing the amount of visual information in the drawing, making it easier to read. To provide bus connections, the Schematic Editor uses a combination of signal and bus **DRAW-WIRE** commands, signal and bus **DRAW-BUS** commands, and the **DRAW-LABEL** command.

A Schematic Editor bus can be divided into two basic components: the bus itself and the signal wire-to-bus interconnections (ARCs). A bus is drawn using the bus (fat) mode of the **WIRE** command. Smooth bus bends (ARCs) are created using the bus (fat) mode of the **BUS** command.

Each signal wire connected to a bus must be individually labeled. As previously discussed, the label must be oriented correctly with respect to the exact end point of the associated signal wire. In addition to a label, a signal (thin) **BUS** can be used to show the signal wire-to-bus interconnection.

It is important to understand that *only signal wire labels* provide the SCHPOST post-processors with connectivity information. Bus wires, fat bus ARCs, and thin signal-to-bus ARCs are cosmetic additions only - they do not supply any information to the SCHPOST post-processors and are used only as *visual aids*.

Drawing buses is easy. The **DRAW-REPEAT** command, discussed in paragraph 3.5, can be used to draw buses very quickly.

Figure 3-9 shows a properly-drawn data bus. Figure 3-10 shows the same bus with only the required SCHPOST post-processor information (no bus lines or arcs are interpreted by SCHPOST). *Both examples produce identical wire/net/pin lists.*

3.3.5 Printing/Plotting SCHDRAW Drawings

SCHDRAW's drawing page files can be printed or plotted. If you have a dot-matrix or laser printer, you can print directly from SCHDRAW. If you are using a plotter, you must exit SCHDRAW and use the SCHPLOT program.

The SCHPLOT program must be used to plot SCHEMA III drawings. You can not plot from SCHDRAW. Plots can be redirected to a file, i.e., spooled to disk, if desired. This feature lets you transport your SCHEMA III drawings to another plotter without having to use SCHEMA III to plot. Chapter 4, *SCHPLOT/SCHPRINT*, contains more information on plotting.

If you are using a printer (dot-matrix or laser), you have many different print modes at your disposal. Depending on what kind of printer you have, these modes affect print resolution, final output size, and the time required to print the drawing. You can print directly from SCHDRAW or exit and use the SCHPRINT program. When printing from SCHDRAW, you can print in foreground or background.

Refer to the SCHDRAW FILE-PRINT command summary in this chapter and Chapter 4 for information on print mode selection. Appendix E, *Hardware Requirements*, contains output size format descriptions and the dot-per-inch print resolutions provided by each printer supported by SCHEMA III.

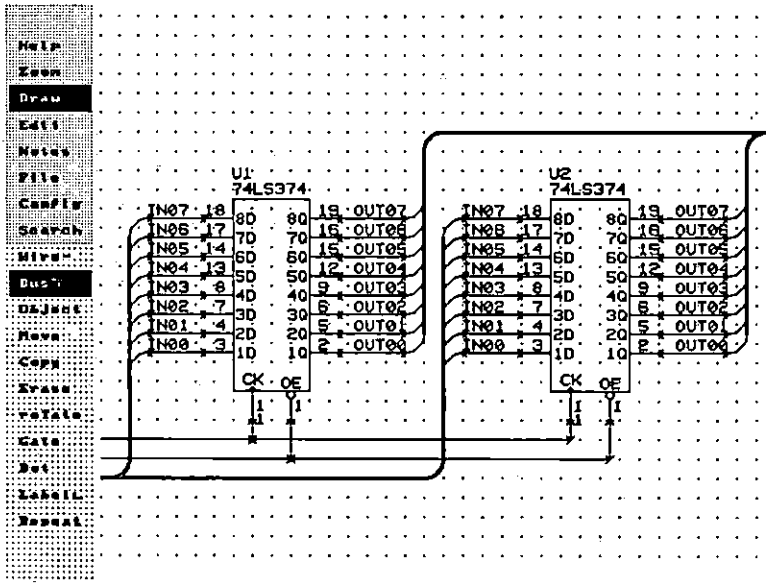


Figure 3-9. Data Bus

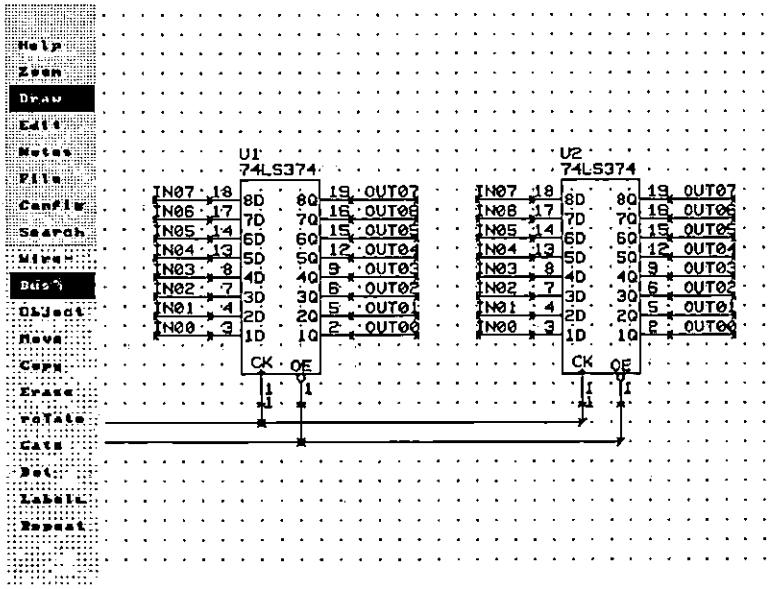


Figure 3-10. SCHPOST Data Bus Components

3.4 USING THE SCHDRAW OBJECT EDITOR

SCHDRAW drawings consist mainly of graphical objects. Schematics contain components (such as ICs, resistors, capacitors) and symbols (such as digital and analog ground). In other types of drawings, like block diagrams and flow charts, objects can be anything from arrowheads to complex geometric shapes with text. SCHDRAW's Object Editor is used to create/edit all these types of objects. The following paragraphs pertain to *schematic component* objects. *It is strongly suggested that you follow through the tutorial in Chapter 1 and read these paragraphs before creating/editing any SCHEMA III schematic component objects!*

3.4.1 Invoking the Object Editor

The Object Editor is invoked by selecting the Schematic Editor FILE-ED OBJ Command Box. Follow the user prompt to invoke the Object Editor. A summary of each Object Editor command is contained in paragraph 3.5.

3.4.2 Types of Objects

SCHEMA schematic component objects are arranged in a tree structure. Each component appearing in the Schematic Editor actually consists of at least two separate objects in the Object Editor: *Name* and *Body*. Name and Body Objects are created/edited in the Object Editor, yet work together to appear as one whole component in the Schematic Editor.

A third type of object, *Nested*, is also used. Nested Objects are always subcomponents of Body Objects.

Figure 3-11 contains a graphic representation of SCHEMA object tree structure. The top-level object, the Name Object, is saved as the actual component name (e.g., 74LS00). Name Objects are literally a few lines of textual information - no graphics are used. Name Objects are created using the NAME Secondary Commands: **NAME-DESC** and **NAME-BODY.**

The **NAME-DESC** Secondary Commands identify the component's company part number (**DESC 1**) and a brief description of the component for the SCHPOST Bill of Materials post-processor (**DESC 2**). **DESC 3** and **DESC 4** are shipped undefined.

The **NAME-BODY** field points to the associated Body Object, which is the component's visual (graphical) object shape. The **BODY** field in the Name Object often points to many different Body Objects. When a component has multiple physical or alternate gate sections (for example, horizontal and vertical resistors, multiple NAND gates, etc.), multiple Body Objects (visual shapes) are required. Body Objects are always named after the **BODY** field with a suffix that ends with **-na**; where **n** is a numeric character representing physical orientation and **a** is an alphabetic character representing gate section. The **n** character can be any numeric value from 1 to 9. The **a** character can be any alphabetical character from A to Z. When objects are retrieved via the Schematic Editor **DRAW-OBJECT** Parameter String, the desired physical orientation and/or gate section is selected by entering **\na** immediately after the component's name. If **\na** is not entered after the component's name, SCHDRAW default-loads the associated Body Object ending with **-1A**. Thus, if only one Body Object is used by a component (no alternate physical orientations or gate sections represented by several Body Objects), save the Body Object with a **-1A** suffix.

Body Objects are the actual object graphics shapes containing visual information such as LINEs, TEXT, Pin Headers (HDRs), etc. All Body

Objects are created using the **BODY** Secondary Commands. For most SCHEMA III components, the Body Object is the lowest level object in the object tree structure. However, for components with commonly-used visual shapes (e.g., NAND gate shapes), a third lower-level object is used to build the Body Object. This object is called a *Nested Object.* Like Body Objects, Nested Objects contain any type of visual information. In theory, Nested Objects can use other Nested Objects. In practice this is seldom done. Nested Objects are retrieved using the **BODY-N OBJ** Secondary Command.

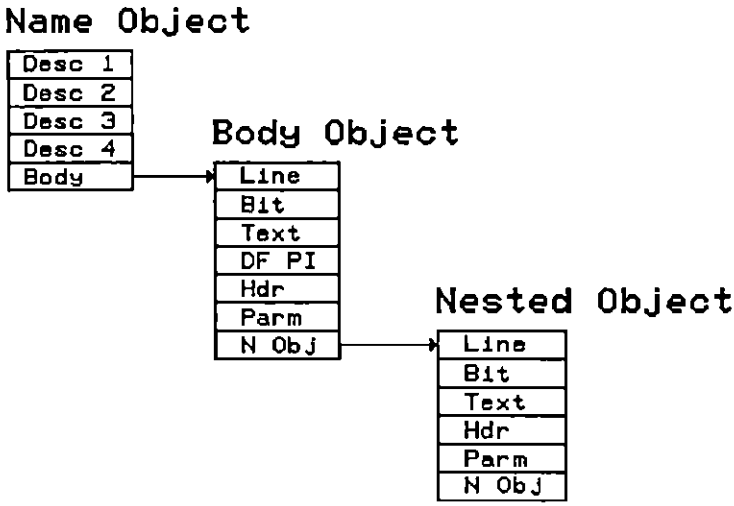


Figure 3-11. SCHEMA III Object Tree Structure

The idea behind Nested Objects is to conserve disk space. Nested Objects can be drawn once and then used over and over again as fundamental building blocks for Body Objects that share the same physical shape but have different pin numbers and other attributes.

Figure 3-12 illustrates the tree structure of SCHEMA III's 74LS00 and 74S00 components. The Name Objects of both the 74LS00 and 74S00 objects contain **NAME-DESC** fields unique to each component (different part numbers and component descriptions). In addition, each Name Object contains an identical **NAME-BODY** field of T00. There are eight T00 Body Objects. Four are used for normal orientations with each gate section (T00-1A to T00-1D) which use ONAND. The remaining four are used for DeMorgan orientations with each gate section (T00-2A to T00-2D) which use ONANDD. Each of these eight Body Objects uses a two common Nested Objects called ONAND and ONANDD (a NAND gate shape) as their fundamental building block.

The following paragraphs discuss Object Editor commands and how they relate to the four object types (Name, Body, Nested and Hierarchical). Detailed descriptions of each Object Editor command are contained in paragraph 3.5.

3.4.3 Name Objects

Name Objects are created/edited using the Object Editor **NAME** Secondary Commands: **DESC** and **BODY**. Name Objects can also be created/edited using the SCHLIB program, discussed in Chapter 6.

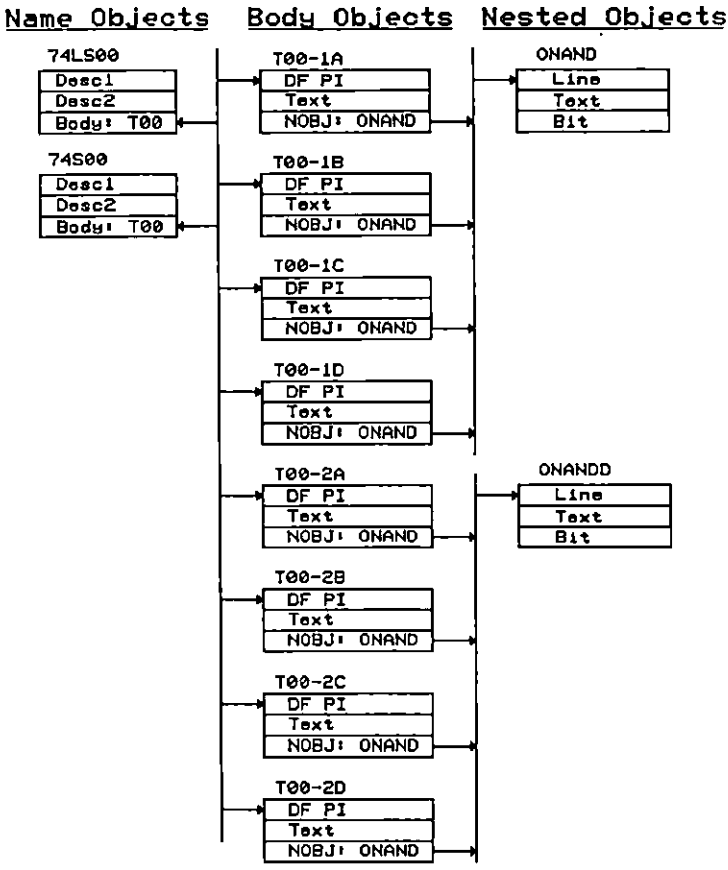


Figure 3-12. SCHEMA III 74LS00/74S00 Tree Structure

The NAME-DESC command is used to create the DESC 1, DESC 2, DESC 3, and DESC 4 component description fields. These fields contain textual data that is printed on SCHPOST's Bill of Materials post-processor output. The DESC fields never appear with the associated component in the Schematic Editor. With SCHEMA III components, the DESC 1 field has been predefined to contain your company's part number for the component. The DESC 2 field should contain a brief description of the component. For most components, the NAME-DESC 1 and NAME-DESC 2 fields literally contain text describing the aforementioned information.

However, discrete components, which may use one Name Object for hundreds of parts (for example a resistor Name Object is used for all resistors regardless of value, tolerance, wattage rating, company part number, etc.), require that some NAME-DESC fields be taken directly from the Schematic Editor DRAW-OBJECT Parameter String entries. This is done using the text:

<Pn>

(where: **n** = 1 to 4) as the NAME-DESC field. This text is then replaced with the nth entry from the DRAW-OBJECT Parameter String and is passed to the SCHPOST Bill of Materials post-processor. For most discrete components, the fourth parameter is your company's part number for the associated discrete component. Thus, the NAME-DESC 1 field is simply:

<P4>

This means that the in-house part number for discrete components must be included as the fourth parameter of the resistor's DRAW-OBJECT Parameter String.

The **NAME-DESC 2** field contains data referencing other **DRAW-OBJECT** Parameter String entries.

A resistor's **NAME-DESC 2** field is:

RESISTOR,<P1>,

This field tells the SCHPOST Bill of Materials post-processor to print the resistor's value (**<P1>**), immediately after **RESISTOR** on the bill of materials output. The actual value is entered as part of the resistor's **DRAW-OBJECT** Parameter String when the resistor is used in the Schematic Editor.

The **NAME-BODY** command contains the prefix of the associated Body Object's name. Although all Body Objects end with a **-na** suffix (such as T00-1A), this suffix is *never* included in the Name Object **NAME-BODY** field. For example, Body Object T00-1A's associated Name Object **NAME-BODY** field is simply: T00. This allows the desired gate section or physical object orientation to be selected from a **DRAW-OBJECT** Parameter String entry - refer to paragraph 3.3.4.2.1.

Finally, all Name Objects are saved as the actual component name, i.e., 74LS00, 6164, 8088, etc. A good example of a Name Object is the 74LS160. From the Object Editor, the 74LS160 Name Object looks like this when **LOAD**ed:

300-0160-001
74LS160 - LS TTL /10 SYNC COUNTER, DIRECT CLEAR
T160

The first line, **300-0160-001,** is the 74LS160's part number assigned using the **NAME-DESC 1** command.

The second line is a brief description of the component assigned with the **NAME-DESC 2** command. These two lines are printed on SCHPOST's bill of materials output everytime a 74LS160 component is used in the Schematic Editor. The third line, **T160**, points to the associated Body Object, T160-1A, which is the actual, visual 74LS160 shape. The T160 field was assigned using the **NAME-BODY** command.

3.4.4 Body/Nested Objects

The **BODY** commands are used to create/edit Body and Nested Objects. Body Objects contain two types of information: visual and post-processor.

To save time and library space, Nested Objects are sometimes used to represent commonly-used graphics shapes or other information. The Object Editor **BODY-N OBJ** command is used to nest objects, i.e., make another object a part of the current object.

The visual information within Body Objects defines what the object will look like when it is used in the Schematic Editor. The **BODY-TEXT, -LINE, -BIT,** and **-HDR** Secondary Commands are used to create this visual information. All component parts, including outline shapes, pin numbers, pin text, pin headers (stubs), etc. are drawn using these commands. An additional command, **BODY-PARM**, defines the physical locations of the **DRAW-OBJECT** Parameter String entries (component name (**PARM N**), reference designator (**PARM R**), and **PARMs 1 - 4**) with respect to the Body Object. The **BODY-PARM** command field locations define physical placement, while the actual parameters are entered in the Schematic Editor **DRAW-OBJECT** Parameter String.

Once a Body/Nested Object is drawn and its parameter (PARM) locations are defined, the visual part of the Body/Nested Object is complete. Next, the pin connection information must be defined for use by SCHPOST's Net/Pin/Wire List post-processors. The **BODY-DF PI**

command defines this information for SCHPOST. **BODY-DF PI** provides three functions: exact pin location, pin type, and pin name.

BODY-DF PI defines pin type and name, while its physical location on the Body Object defines exact pin location. **BODY-DF PI** operation is similar to the Schematic Editor **DRAW-LABEL** command, except **DF PI** entries are not displayed with the Body Object when it appears in the Schematic Editor.

The pin definitions on hierarchical objects should match net labels on the page that corresponds to the object. When the schematic is flattened, the signals going into the pins of the hierarchical object are actually going into the nets on the corresponding page. For further discussion on hierarchical objects see paragraph 3.4.5.

BODY-DF PI text consists of a one-character alphabetical pin type followed by a one to three-character pin name (or number). Once entered, the **BODY-DF PI** text is placed at the exact end of the applicable pin header (**BODY-HDR**) or stub in a manner similar to assigning **LABEL**s to signal wires in the Schematic Editor. The **BODY-DF PI** command description in paragraph 3.5 outlines legal **BODY-DF PI** field formats. Note that **BODY-DF PI** can only be used with Body Objects. *Do not assign pin definitions to Nested Objects.*

BODY-PIN automates the entire pin creation process and can simplify the creation of pin. It essentially makes a region of **HDR**s, **TEXT**, and **DF PI**.

3.4.4.1 Assigning Power and Ground

The **BODY-DF PI** command is also used to assign power and ground to Body Objects. The *power and ground assignment lines* identify the component's power and ground pins for the SCHPOST post-processors

and are not displayed with the object when it appears in the Schematic Editor. Figure 3-13 shows a typical digital Body Object. The power/ground assignment lines are:

I16 | @+5 and I8 | @GND

Chapter 1 and the **DF PI** command description in paragraph 3.5 outline the procedure to assign power and ground to Body Objects.

The Alternate method would be to select **BODY-PIN** until the '@' icon is displayed. Then type : **I16,+5** or **I8,GND** to automatically create these hidden pins.

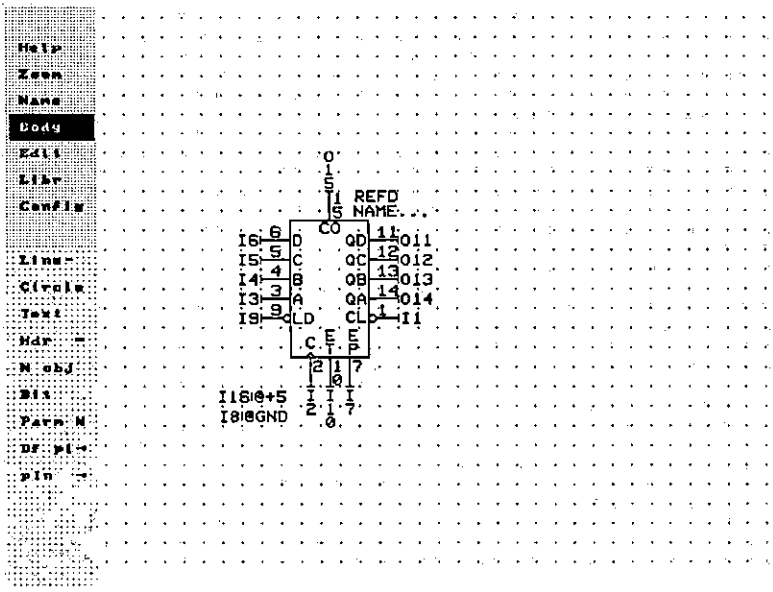


Figure 3-13. Typical Digital Body Object

3.4.4.2 Referencing Parameters Within BODY-PIN Entries

The Object Editor lets you reference Schematic Editor **DRAW-OBJECT** Parameter String entries within **BODY-PIN** assignments.

For example: Select the '@' icon for **BODY-PIN** command. The power and ground assignment lines for an Op Amp Body Object could be by typing in the Text window:

I16,<P1> <Enter> and I8,<P2> <Enter>

where **<P1>** and **<P2>** are the first and second Schematic Editor **DRAW-OBJECT** Parameter String parameter entries. Thus, if the Op Amp's component's name is OPAMP, the corresponding Schematic Editor **DRAW-OBJECT** Parameter String might be:

OPAMP,U2,+12,-12 <Enter>

NAME,REFD,<P1>,<P2>

where +12 is **<P1>** and -12 is **<P2>**. The idea behind referencing parameters within **BODY-PIN** entries is to allow you to assign pin information to components "on the fly", as you design. This is an excellent feature for components whose power and ground assignments vary, depending on the design application. This feature also applies to connector objects.

3.4.4.3 Making Connectors

Connector objects are really no different than other objects. Connectors are drawn and assigned pin definitions using the **BODY-DF PI** or **BODY-PIN** command, just like other Body Objects. Connector Body Objects can be drawn three ways, however.

The first method is to draw the connector as a single, complete object. For example, a DB25 connector may be drawn the same way as a 25-pin IC with a single Name Object and Body Object. The Body Object would have 25 pin headers (**BODY-HDR**), with all 25 pins individually defined (**BODY-DF PI**) or (**BODY-PIN**).

The second and third methods relate to how connectors are sometimes shown in schematic diagrams.

One connector is often shown in several different places throughout a schematic, with each location showing different pin number assignments. With this method the aforementioned DB25 connector could be shown as 25 individual objects. A single Name Object would be created, followed by 25 separate Body Objects, each saved with -1A through -1Y suffixes. This technique resembles the method used by Omation to represent multiple gate sections of an IC, such as the 74LS00.

The third method is considerably easier. By referencing a parameter in the connector's Body Object **BODY-DF PI** entry, a single Body Object can be created to represent each pin of the DB25. For example, a single Body Object with one pin is created. The pin's **BODY-DF PI** entry is:

I<P1>

Assume the corresponding Name Object is DB25. From the Schematic Editor **DRAW-OBJECT** command:

DB25,J1,1 <Enter>

retrieves the DB25 Body Object with an assigned reference designator of J1 and assigns pin 1 to the pin. The next time DB25 is used the Schematic Editor **DRAW-OBJECT** Parameter String entry could be:

DB25,J1,2 <Enter>

This retrieves the same Body Object, but assigns pin 2 to the pin.

A problem can arise with this method because the multiple use of one component with the same reference designator creates an error output from SCHPOST's post-processors. To circumvent this problem, always name connector Name Objects created with this method beginning with two caret (^ ^) characters. For example: ^ ^**DB25**. When two caret characters are used, SCHPOST will not output the multiple use error message for the component.

The end result is one connector object with one reference designator used over and over again in a schematic, with a large number of different pins used.

The component ^ ^CONN is a sample connector object using this scheme. The two ^ ^CONN Body Object orientations, **APIN-1A** and **APIN-2A**, show a pin referencing Parameter 1.

3.4.5 Hierarchical Objects

Another way to save time is to use Hierarchical Objects. Hierarchical Objects are created to represent a low-level schematic drawing which is used several times within other drawings. When drawing a top level hierarchical schematic, a Hierarchical Body Object can be loaded multiple times and connected quickly and efficiently. The hierarchical schematic that represents the hierarchy *must be in the same sub-directory* as the library that contains the hierarchical object (It could also be in the current directory). Several Hierarchical Body Objects can be used in a schematic drawing. There is no limit to the number of drawings (hierarchical objects) that can be used to complete the top level hierarchical schematic.

The Object Editor **BODY** commands are used to create/edit Hierarchical Body Objects just as Body and Nested Objects are created. When defining the pins using the **BODY-DF PI** or **BODY-PIN** command, the pin definitions on the hierarchical object should match the net labels on the lower-level schematic. This allows the post-processor to flatten the high-level schematic and produce a net list showing the signals going into the pins of the hierarchical object that are really going into the nets on the corresponding hierarchical schematic. Refer to Chapter 7, *SCHPOST*, for detailed information regarding the post-processor functions.

Name Objects are also used for hierarchical objects to identify the part number and describe the drawing. Hierarchical Name Objects are created using the **NAME** command in the Object Editor. The Hierarchical Name Object name for all hierarchical objects *must have a corresponding hierarchical schematic of the same name*, that shows the logic of the hierarchical object.

The Hierarchical Drawing section in Chapter 1 (1.5.4.6) outlines the procedures to produce Hierarchical Body Objects, Name Objects and Drawings.

3.4.6 Maximum Body Object Size

The maximum allowable size of a Body Object including all text is as large as the Object Editor screen will allow. However, if the object is greater than 900 square visual grid units, then the user will have to go to a higher zoom level to place the part. This is similiar to what happens with regions that are too large.

3.4.7 Object Naming Conventions

All schematic component Name, Body, Nested and Hierarchical Objects included with SCHEMA III use naming conventions. Your objects do

not have to follow these conventions but since they have already been predefined for you, it may be most productive to use them. Refer to Appendix F for a description of the object naming conventions used with SCHEMA III's component objects.

The maximum length of any object name (Name, Body, Nested) including extension, is eight characters.

The Name Object of the Hierarchical Body Object is limited to 5 characters including #H (Do not include the orientation/gate section suffix). This is executed using the NAME-BODY command.

When saving the Hierarchical Body Object using the LIBR-SAVE command, all Body Object names for hierarchical objects must begin with #H.

3.5 COMMAND DESCRIPTIONS

This paragraph contains detailed descriptions of each SCHDRAW command. Commands are listed alphabetically and described as follows:

- **Editor:** Schematic, Object, or both
- **Use:** Defines reason for use
- **Function:** Explains what the command does
- **Operation:** Lists procedure to use command
- **See Also:** Identifies related commands/information, if applicable
- **Notes:** Identifies special conditions, limitations, etc. associated with command, if applicable.

BODY

Editor: Object

Use: To create/edit Body, Nested and Hierarchical Objects.

Function: Provides access to **BIT, CIRCLE, DF PI, HDR, LINE, N OBJ, PARM, PIN,** and **TEXT** Secondary Commands.

Operation: When selected, the **BODY** Command Box lights and **BIT, DF PI, HDR, LINE, N OBJ, PARM, PIN,** and **TEXT** Secondary Commands appear in the lower portion of the Command Window.

BODY-BIT

Editor: Object

Use: To create or edit images dot-by-dot (pixel-by-pixel).

Function: Allows bit image creation and editing.

Operation: When selected, the **BIT** Command Box lights. An icon in the **BIT** Command Box indicates the bit mode (**1, 2,** or **3**).

The **BIT 1** mode allows for the selection of a previously-drawn bit image for editing. The **BIT 2** mode allows bit editing and the **BIT 3** mode allows the edited bit image to be placed in the Drawing Window.

To edit a bit image currently displayed in the Drawing Window, first deactivate the cursor grid (**C GRID** Command Box extinguished). Next,

select **BIT 1** and position the cursor over the bit image. Now click the mouse button or press **<Ins>** twice. The **BIT** Command Box increments to **BIT 2** and the Bit Window appears with a pixel-level view of the selected bit image. (To create a new bit image (when in **BIT 1**), simply reselect the **BIT 1** Command Box. The command box increments to **BIT 2** and the empty Bit Window is displayed.).

To edit in the **BIT 2** mode, position the cursor on the desired Bit Window location. Click the mouse button or press **<Ins>** twice to add pixels. To erase a pixel, position the cursor over the pixel and click the mouse button or press **<Ins>** twice. When finished with the bit image, select the **BIT 2** Command Box. The Bit Window is replaced with the Drawing Window and **BIT 3** is displayed in the BIT Command Box. If an existing bit image was being edited, that image is erased and will no longer be visible. Position the cursor in the Drawing Window and press and hold the mouse button or press **<Ins>**. The newly-created (or edited) bit image appears in the Drawing Window and can be moved throughout the Drawing Window as long as the mouse button is held pressed down. The Drawing Window automatically pans if the bit image is moved to any edge of the screen. Move the bit image to the desired location and release the mouse button or press **<Ins>**. Multiple copies of the same bit image are drawn by repeatedly pressing the mouse button or **<Ins>**.

Notes: The Bit Window displays a 68 (X) by 40 (Y) pixel bit map.

When multiple bit images are added to the Drawing Window, each represents a unique image. Thus, if one of the images is edited, changes apply only to the edited image - none of the other images are changed.

BIT is used to either create new bit images or edit predrawn bit images. BIT cannot be used to edit LINEs, HDRs, DF PI, TEXT, etc.

BODY-CIRCLE

Editor: Object

Use: To add circles and arcs to objects.

Fuction: Symbolizes thin circles and arcs for Body, Nested, and Hierarchical Objects.

Operation: When selected the CIRCLE Command Box lights and the Text Window is displayed. The circle/arc command line format is:

RADIUS <Enter> (Draws circle of specified pixel radius)

RADIUS,BEGIN,END <Enter> (Draws arc from specified beginning point to ending point).

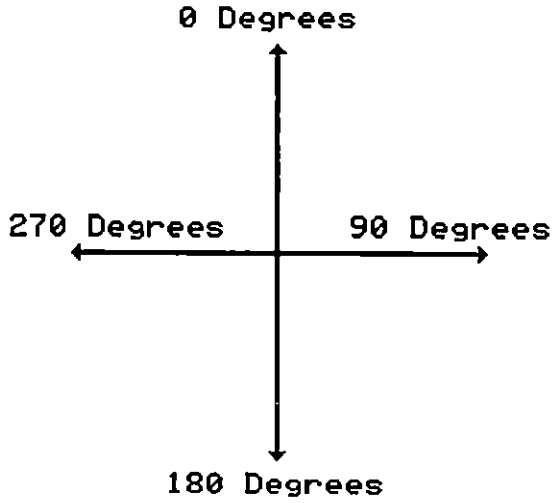
where **RADIUS** is the desired circle or arc radius in pixels; **BEGIN** is the arc beginning point in degrees, specified in 1-degree increments; **END** is the arc end point in degrees, specified in 1-degree increments. The minimum radius allowed is 2 pixels and the maximum is 256 pixels. For example, to draw a circle with a radius of 32 pixels, enter:

32 <Enter>

To draw an arc from 90 degrees to 270 degrees, with a radius of 32 pixels, enter:

32,90,270 <Enter>

After <Enter> is pressed, position the cursor in the Drawing Window. Next, either press and hold the mouse button or press <Ins>. The circle/arc appears and can be moved anywhere in the Drawing Window. Release the mouse button or repress <Ins> to place the circle/arc.



BODY-DF PI

Editor: Object

Use: To assign pin definitions and locations to Body and Hierarchical Objects for use by the SCHPOST post-processors.

Function: Displays the desired pin definition in the Drawing Window.

Operation: When selected, the **DF PI** Command Box lights and the Text Window appears. An icon in the **DF PI** Command Box indicates the currently-selected pin definition physical orientation - to the left of (above) or right of (below) the associated pin header (**BODY-HDR**). Select the **DF PI** Command Box until the desired orientation is selected. Then, enter the pin definition in the Text Window followed by <Enter>. Position the cursor in the Drawing Window and press and hold the mouse button (or press **<Ins>**). The pin definition appears in the Drawing Window with a small connection dot (or bar, if **CONFIG-WI HDR** is enabled). The pin definition can be moved throughout the Drawing Window as long as the mouse button is held pressed down. The Drawing Window automatically pans if the pin definition is moved to any edge of the screen. Move the pin definition to the desired pin header/line, align the connection dot/bar with the exact pin header/line end point, and release the mouse button (or press **<Ins>**). The Text Window reappears and another pin definition may be entered. The pin definition format is:

Axxx

where: A is the pin type (interpreted by SCHPOST's Design Rule Check post-processor) and xxx is the pin name defined by the user (ignored by post-processing).

Legal A values are:

I	Input
O	Output
T	Tri-state Output
B	Bidirectional
C	Open Collector
@	Absolute (next characters are a signal label)
X	Don't Care (pin not involved in design rule checks).

Legal xxx values encompass any combination of three user-defined alpha and/or numeric characters. The xxx value is normally assigned the component's pin number.

DF PI is also used to assign power and ground to components. Power and ground connections are assigned with the 'at' (@) symbol. The @ symbol informs the SCHPOST post-processors that the associated characters define an 'Absolute' label assigned to the applicable location. To assign power or ground to a pin, first define the pin. For example: I8. Then place the definition anywhere in the Drawing Window. Next, reselect the DF PI Command Box to display the opposite DF PI orientation. Then, enter: @ followed by GND (or +5, -15, etc.). Position the @GND connection dot (or bar if CONFIG-WI HDR is active) on top of the I8 connection dot (or bar). This assigns ground to pin I8. For example:

```
I8 | @GND
```

PARMs can be referenced within the DF PI entry. This is quite useful when connector objects are created. For example, a DF PI entry could be:

```
I<P1>
```

This defines a pin header type as an input (I) with the actual pin number taken from the Parameter 1 (**<P1>**) entry in the object's Schematic Editor **DRAW-OBJECT** Parameter String. This allows the same connector object to be used many times in the same drawing with different pin number assignments.

PARMs can also be referenced in power and ground assignments. This is most useful with analog components. For example, a typical analog component power assignment line is:

I7 | @VSS

which could be changed to:

I7 | @<P1>

allowing the desired voltage to be specified as the Parameter 1 entry in the object's Schematic Editor **DRAW-OBJECT** Parameter String.

Some additional pin definition formats are:

T14 - Assigns horizontal pin header with a tri-state pin type and pin name of 14

See Also: **CONFIG-WI HDR and BODY-PIN**

Notes: **DF PI** must only be used with Body Objects. Do not use **DF PI** within *Nested* Objects.

To prevent Routing Errors during post processing, the **C Grid** (Cursor Grid) should be enabled. This will insure that when wires are connected to pin definitions, the two will be aligned.

Pin definitions are used by SCHPOST's Design Rule Check post-processor. The values assigned to pin types are used on digital components. Other components such as transistors, resistors, capacitors, etc. are assigned X (Don't Care) for all pin type definitions. Connector pins require correct pin types to ensure the Design Rule Check post-processor does not find floating inputs or multiple output errors. The pin name value is not interpreted by the SCHPOST post-processor - it is just printed on wire lists and net lists. Thus, names (E, CLK) or numbers (1, 5) are suitable for use as pin name values.

The pin definitions on hierarchical objects should match net labels on the page that corresponds to the object. When the schematic is flattened, the signals going into the pins of the hierarchical object are actually going into the nets on the corresponding page.

Like **LABEL**s, it is much easier to properly align **DF PI**s when **CONFIG-WI HDR** is active.

DF PI entries are not displayed with components in the Schematic Editor.

DF PI entries should be in uppercase. Lowercase entries will be automatically converted to uppercase when the SCHPOST function is invoked. *For hierarchical objects, SCHPOST will not automatically convert lowercase to uppercase.*

BODY-PIN will automate the pin creation process.

BODY-HDR

Editor: Object

Use: To add pin headers (stubs) to a Body, Nested or Hierarchical Object.

Function: Displays vertical and horizontal pins (with and without inverter bubbles).

Operation: When selected, the **HDR** Command Box lights and an icon in the **HDR** Command Box displays the currently-selected pin header type. Six types are available: vertical, horizontal, inverter bubble at left (or top), and inverter bubble at right (or bottom). Select the **HDR** Command Box until the desired pin header type is selected. Position the cursor in the Drawing Window and press and hold the mouse button (or press **<Ins>**). The pin header appears and can be moved throughout the Drawing Window as long as the mouse button is held pressed down. The Drawing Window automatically pans if the header is moved to any edge of the screen. Move the header to the desired location and release the mouse button or press **<Ins>**. Once **HDR** is selected, there is no limit to the number of pin headers that can be added to objects. To add another pin header, press and hold the mouse button (or press **<Ins>**), position the cursor in the Drawing Window, and release the mouse button or press **<Ins>**.

Notes: Pin headers are used to quickly draw an object's pin stubs - they carry no post-processing information. Thus, the **BODY-LINE** command can also be used to draw pin stubs if a different pin stub length is desired.

A limited amount of library disk space (approximately 5 bytes per header) can be saved by adding similar pin headers starting from a left

direction and moving right (for vertical headers) or from top to bottom (horizontal headers).

BODY-LINE

Editor: Object

Use: To add orthogonal thin, thick, dotted, and dashed lines to objects. When CONFIG-DIAG is selected, allows thin diagonal lines to be drawn.

Function: Symbolizes thin and thick lines, dotted, and dashed orthogonal lines for Body, Nested and Hierarchical Objects.

Operation: When selected, the LINE Command Box lights. Icons within the LINE Command Box display the currently-selected line type (thin, thick, dotted, dashed, circle/arc).

To draw orthogonal lines, ensure the CONFIG-DIAG Command Box is extinguished. Select the LINE Command Box until the desired line type is displayed. Position the cursor in the Drawing Window. Press and hold the mouse button (or press <Ins>) and move the cursor to draw a line. The first cursor movement determines line direction (X or Y). Elbow (L-shaped, 90-degree) lines are easily drawn. Move the cursor up and the line appears to move up the Drawing Window. Move cursor left or right and the line routes left or right, respectively. As long as the mouse button is held down, line length and direction (in both X and Y planes) can be adjusted. If a line is drawn in the wrong direction, reverse cursor movement to reverse direction. The Drawing Window automatically pans if a line is routed to any edge of the screen. After the desired line is drawn, either release the mouse button or press <Ins>.

Once LINE is selected, there is no limit to the number of lines that can be added to an object. To add another line, press and hold the mouse button (or press <Ins>) and move the cursor in the desired direction.

To draw diagonal lines, ensure the CONFIG-DIAG Command Box is lighted. Next, select the LINE Command Box until the solid thin line icon is displayed. Position the cursor in the Drawing Window. Press and hold the mouse button (or press <Ins>) and move the cursor to draw the diagonal line. Place the line as described above.

See Also: CONFIG-DIAG

Notes: All BODY-LINEs are identical. Thus, a line can be used to represent a component body, pin header, bar over a signal name, etc. EDIT-LINE can be used to modify the line.

Diagonal lines are available in the solid thin LINE mode only.

BODY-N OBJ

Editor: Object

Use: To add a Nested Object to the current Body/Nested Object.

Function: Displays Nested Object in the Drawing Window.

Operation: When selected, the N OBJ Command Box lights and the Text Window appears. Enter the desired preexisting Nested Object name followed by <Enter>. Position the cursor in the Drawing Window and press and hold the mouse button (or press <Ins>). The

Nested Object appears and can be moved throughout the Drawing Window as long as the mouse button is held pressed. The Drawing Window automatically pans if the object is moved to any edge of the screen. Move the object to the desired location and release the mouse button or press **<Ins>**.

Library directories can be viewed through **N OBJ**. Refer to the Schematic Editor **DRAW-OBJECT** command for additional information.

See Also: DRAW-OBJECT

Notes: **N OBJ** uses a pointer to the actual Nested Object. For example, the T00-1A Body Object contains a Nested Object named ONAND, a graphic representation of a NAND gate. When T00-1A is called, SCHDRAW automatically retrieves the current ONAND Nested Object to create the T00-1A NAND gate shape. If ONAND is changed, T00-1A reflects the ONAND change. Thus, everytime T00-1A is used, the current ONAND Nested Object is also used.

BODY-PARM

Editor: Object

Use: To assign parameter locations to a Body, Nested or Hierarchical Object.

Function: Displays current parameter replacement text in the Text and Drawing Windows.

Operation: When selected, the **PARM** Command Box lights and the Text Window appears. An icon in the **PARM** Command Box displays

the currently-selected parameter type: N (name), R (reference designator), or 1-4. Select the **PARM** Command Box until the desired parameter type is selected. As the command box is selected, the currently-assigned parameter replacement text, if defined, is displayed in the Text Window. To add new parameter replacement text or modify existing text, enter the new parameter replacement text in the Text Window followed by **<Enter>**. The Text Window disappears. Position the cursor in the Drawing Window and press and hold the mouse button or press **<Ins>**. The desired parameter replacement text appears in the Drawing Window. It can be moved throughout the Drawing Window as long as the mouse button is held pressed down. The Drawing Window automatically pans if the text is moved to any edge of the screen. Move the parameter replacement text to the desired location and release the mouse button or press **<Ins>**.

See Also: EDIT-PM OBJ, DRAW-OBJECT

Notes: **PARM** text only defines the *physical location* for the actual parameter values entered as part of the Schematic Editor's **DRAW-OBJECT** Parameter String.

PARM N is normally entered as **NAME....** and positioned near the Body Object where you want the component's name to appear in the Schematic Editor. Since **NAME....** is literally replaced with the actual name of the associated Name Object in the Schematic Editor, be sure to include enough periods after **NAME** to pad the **PARM N** length to equal the length of the associated Name Object name. For example, if you create a 74HC4050 Name Object, the Body Object's **PARM N** entry should be:

NAME....

or *eight total characters* (NAME plus four periods to equal **74HC4050**). Thus, **NAME....** gives SCHEMA III enough room to display **74HC4050** in the Schematic Editor.

PARM R is normally expressed as **REFD** and should be positioned near the Body Object where you want the component's reference designator to be displayed in the Schematic Editor. **REFD** is then literally replaced with the component's reference designator (whether it is auto-assigned by the Schematic Editor or entered as part of the **DRAW-OBJECT** Parameter String) when the component is used in the Schematic Editor.

PARMs 1-4 are normally used with analog and discrete components to position the part value (such as 10K, 100uF, etc.), power rating (such as 1W), tolerance (1%, 5%, 10%, etc.), or any other information you want displayed near the component in the Schematic Editor. Ensure the length of your **PARM 1-4** assignments are adequate to match the length of the actual **DRAW-OBJECT** Parameter String entries. The Schematic Editor **EDIT-PM OBJ** command can be used to display these **PARMs** when the component is used in the Schematic Editor.

BODY-PIN

Editor: Object

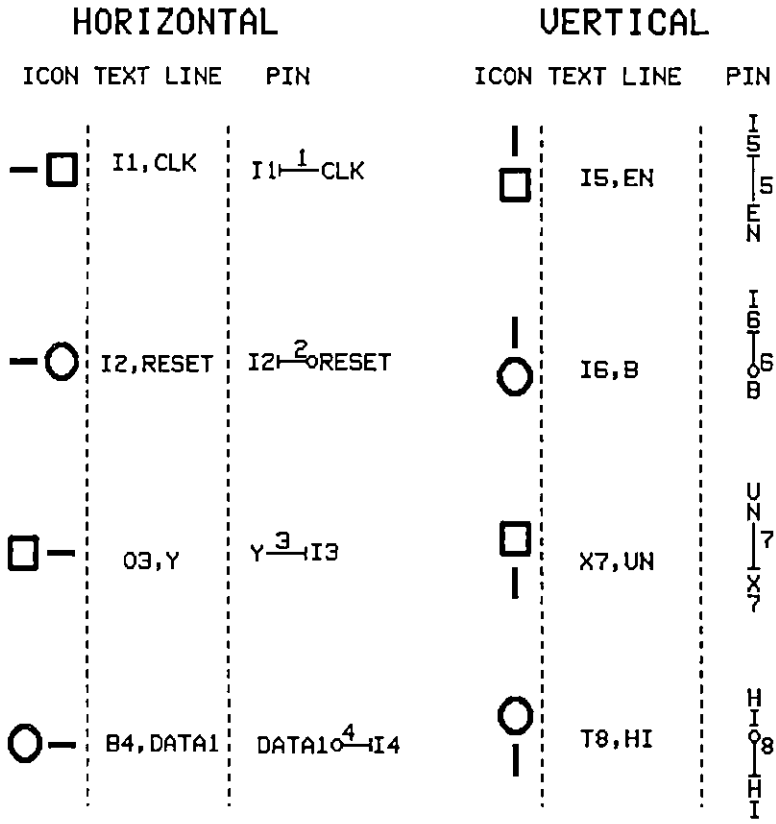
Use: To automate the pin creation process.

Function: Allows the user to type in the pin info and have the header, body text , and pin defintion all properly spaced and ready for placement.

Operation: This command has icons that are either normal or bubbled and horizontal or vertical. Either the bubble or the square box indicate which end is attached to the body of the part. The header is displayed with the pin definiton displayed. The Body text is given off grid and centered on the pin and the pin text shows the pin value. Precede the pin value with the attribute, **I, O, X, C, B,** or **T.** If you want Body text to be given with the pin then type a delimiting ',' followed by the string of text. If you are creating a hierarchical part the pin definitions are not pins, but net names. Thus, the first letter of the pin definition is not the attribute. Therefore, use a **<Space>** as the attribute value.

The PIN command actually creates a region and places in that region the text , header, and definiton for placement. However once you place them, they are separte entities. If you misplace a pin and immediately select the **EDIT-REGION** command you will see the region box encloses the pin and you can select the **MOVE** command and place the region.

The following figure gives examples of the pins.


HORIZONTAL
ICON TEXT LINE
PIN
I1,CLK
I1 1 CLK
I2,RESET
I2 2 RESET
O3,Y
Y 3 I3
B4,DATA1
DATA1 4 I4
VERTICAL
ICON TEXT LINE
PIN
I5,EN
I5 5 EN
I6,B
I6 6 B
X7,UN
UN 7 X7
T8,HI
HI 8 HI

BODY-TEXT

Editor: Object

Use: To add text to Body, Nested and Hierarchical Objects.

Function: Allows upper or lowercase medium text (only) in various orientations to be added to objects.

Operation: This command operates identically to the NOTES-TEXT command except it uses medium text only.

See Also: NOTES-TEXT

Notes: The text will appear in upper or lowercase as typed.

CONFIG

Editor: Schematic/Object

Use: To set SCHDRAW configuration parameters.

Function: Provides access to **C GRID, COLORS, DIAG, LIBR, PAN, RPT, SAVE, V GRID, WI HDR,** and **XY WDW** Secondary Commands.

Operation: When selected, the **CONFIG** Command Box lights and **C GRID, COLORS, DIAG, LIBR, PAN, RPT, SAVE, V GRID, WI HDR,** and **XY WDW** Secondary Commands appear in the lower portion of the Command Window.

Notes: When SCHDRAW is initialized, all **CONFIG** parameters default to values assigned to the current drawing's configuration file variables.

CONFIG-C GRID

Editor: Schematic/Object

Use: To activate or deactivate the invisible cursor grid.

Function: Snaps cursor movement to an invisible, 8-pixel grid mapped on the Drawing Window.

Operation: When selected, the **C GRID** Command Box lights and an invisible cursor grid covers the Drawing Window. Reselect the **C GRID** Command Box to remove the grid from the Drawing Window and extinguish the **C GRID** Command Box.

See Also: CONFIG-V GRID

Notes: C GRID can be selected by pressing the <F10> Function key.

The invisible cursor grid facilitates object and text alignment. Do not attempt to draw schematics without C GRID selected because the SCHPOST post-processors may not interpret connections made without the cursor grid.

When C GRID is on, the last two digits of the Cursor Coordinate Window will always be .00 or .08 (cursor movement restricted to eight pixels per movement). When both C GRID and V GRID are activated, cursor movement occurs on or half-way between the visible 16-pixel grid dots.

CONFIG-COLORS

Editor: Schematic

Use: To modify the colors of SCHDRAW online.

Function: Alters color scheme.

Operation: When selected the COLORS Command Box lights. The current Hex numbers representing the current color configuration are displayed. The order is as follows:

DRAWING BACKGROUND,
CURSOR/IMAGE,
BORDERS,
DRAWING COLOR,
PRIMARY MENU BACKGROUND
PRIMARY MENU FOREGROUND
SECONDARY MENU BACKGROUND,
SECONDARY MENU FOREGROUND

Enter the new values and press **<Enter>**.

See Also: Chapter 5, *SCHCFG,* **GRAPH=** for more information on how to derive colors.

CONFIG-DIAG

Editor: Schematic/Object

Use: To activate or deactivate the diagonal LINE feature.

Function: Allows diagonal LINEs to be drawn.

Operation: When selected, the DIAG Command Box lights. When NOTES-LINE (Schematic Editor) and BODY-LINE (Object Editor) are used, thin diagonal lines (any length, any angle) can be drawn.

See Also: NOTES-LINE, BODY-LINE

Notes: CONFIG-DIAG can also be selected by pressing the **<F4>** function key.

CONFIG-LIBR

Editor: Schematic/Object

Use: To display the names of the current object libraries assigned to the drawing's associated configuration file LIBR variables.

Function: Displays all currently-configured object library names in the Text Window.

Operation: When selected, the LIBR Command Box lights, an icon in the LIBR Command Box displays the currently selected object library number, and the library name is displayed in the Text Window. If you wish to alter the library in use, type in the new library and press <Enter> to complete the change. If you wish to use Browse to bring up a list of libraries, type in the path and use the wildcard characters, '*' or '?'. A list of the libraries will be displayed. Use the mouse or the cursor keys to select the library. Reselecting the LIBR Command Box displays the next object library in the Text Window. Only libraries assigned to the drawing's configuration file LIBR variables are displayed. The SAVE command will then make any permanent changes to the configuration file.

CONFIG-PAN

Editor: Schematic/Object

Use: To adjust the Drawing Window panning speed.

Function: Sets Drawing Window panning X and Y speed values.

Operation: When selected, the **PAN** Command Box lights and displays a **1, 2,** or **3.** The value **2,** is the default. Users who are using hi resolution graphics adapters (1024x768) may want a quicker pan and would select the value **3.** Users who are working with a CGA mode will want to select **1.**

CONFIG-RPT

Editor: Schematic/Object

Use: To set the X and Y repeat offsets for images repeated using the **DRAW-REPEAT** command (Schematic Editor) or **<F9>** Function key (Schematic/Object Editors).

Function: Determines stepping offsets between images in the Schematic and Object Editors.

Operation: When selected, the **RPT** Command Box lights and the Text Window appears with the currently assigned repeat offset value. An icon within the **RPT** Command Box displays the currently-selected repeat offset type (**X** or **Y**). Select the **RPT** Command Box until the desired repeat offset type is selected. Edit or enter a numeric offset value corresponding to the number of pixels desired between repeated images. A positive offset value for **X** places the repeated image(s) to the right of the original image. A positive offset value for **Y** places the repeated image(s) below the original image. A negative value for **X** places the repeated image(s) to the left of the original image. A negative value for **Y** places the repeated image(s) above the original image. After the desired offset has been entered, press **<Enter>**. The **RPT** Command

Box icon automatically toggles to the other offset type. Enter the value for the other offset (followed by **<Enter>**) or ignore.

See Also: DRAW-REPEAT

CONFIG-SAVE

Editor: Schematic/Object

Use: To save configuration changes to the drawing config file.

Function: Save the current configuration of SCHDRAW.

Operation: Select **SAVE**, and the Text window displaying the currently loaded config path and file. Alter the path or name if necessary and then press <Enter>. The parameters saved to the **DRAWING.CFG** are: **LIBR1-LIBR9=, RPTX/Y=, XY=, CGRID=, VGRID=, WIHDR=, DIAG=,** and **VLABEL=.** The parameters **PAN=** and **COLORS** are saved to the main config file, **SCHEMA.CFG.**

CONFIG-VLABEL

Editor: Schematic/Editor

Use: To activate the vertical icons for **DRAW-LABEL** and **BODY-DF PI.**

Function: Activates the vertical icons for creating vertical labels.

Operation: When selected, the **VLABEL** Command box lights and the vertical icons can be selected for the **DRAW-LABEL** and **BODY-DF PI** command.

See Also: CONFIG-VLABEL

Notes: If you normally do not use vertical labels, then you can de-activate this command within the SCHEMA or DRAWING Configuration files.

CONFIG-V GRID

Editor: Schematic/Editor

Use: To display or remove the Drawing Window Visual Grid.

Function: Displays the Visual Grid in the Drawing Window or removes the grid.

Operation: When selected, the **V GRID** Command Box lights and the Visual Grid covers the Drawing Window. Reselect the **V GRID** Command Box to remove the grid from the Drawing Window and extinguish the **V GRID** Command Box.

See Also: CONFIG-C GRID, CONFIG-XY

Notes: When **V GRID** is selected or deselected, the cursor disappears for several seconds while the Visual Grid is added or removed from the Drawing Window.

V GRID is not associated with the cursor grid.

The distance between V GRID dots is 16 pixels. When C GRID is selected, cursor movement is restricted to on or half-way between the V GRID dots.

CONFIG-WI HDR

Editor: Schematic/Object

Use: To identify signal wire beginning and end points, label assignment points, and component pin header connection points to simplify schematic connections.

Function: Redraws the current drawing with Wire Headers (arrowheads) at signal wire beginning and end points, and vertical or horizontal identification bars at label assignment and component pin header connection points.

Operation: When active, the WI HDR Command Box is lighted and Wire Headers are displayed at applicable beginning and end points of wires, labels, and component pin headers. When WI HDR is not active, the WI HDR Command Box is extinguished and Wire Headers are not visible.

Notes: Wire Headers ensure proper identification of signal wire label assignments, and wire-to-wire and wire-to-component connections. WI HDR should be active when drawing schematics.

When WI HDR is selected or deselected, the cursor disappears for a few seconds while the current drawing is redrawn.

CONFIG-XY WDW

Editor: Schematic

Use: To toggle On/Off the Cursor Coordinate Window.

Function: To invoke or remove the Cursor Coordinate window from the Drawing Window.

Operation: When selected the XY WDW Command Box will either light or turn off.

See Also: Chapter 5, *SCHCFG* , Drawing Parameters Configuration Menu

DRAW

Editor: Schematic

Use: To add schematic-related graphics such as wires, arcs, objects, labels, dots, etc. to the current drawing.

Function: Provides access to **BUS, COPY, DOT, ERASE, GATE, LABEL, MOVE, OBJECT, REPEAT, ROTATE,** and **WIRE** Secondary Commands.

Operation: When selected, the **DRAW** Command Box lights and **BUS, COPY, DOT, GATE, LABEL, MOVE, OBJECT, REPEAT, ROTATE,** and **WIRE** Secondary Commands appear in the lower portion of the Command Window.

Notes: Most **DRAW** operations are interpreted by SCHPOST's post-processors. However, if post-processing is not used (non-schematic drawing), **DRAW** Secondary Commands can be used to create any desired graphics without regard to post-processor-imposed drawing conventions.

DRAW-BUS

Editor: Schematic

Use: To provide signal-to-bus interconnections and bus right-angle (smooth 90-degree) corners.

Function: Symbolizes signal or bus wire bends with thin or fat curved graphics.

Operation: When selected, the **BUS** Command Box lights and icons within the **BUS** Command Box display the currently-selected arc orientation. Four unique **BUS** orientations are available by selecting the **BUS** Command Box. Arc thickness is determined by the setting of wire thickness in the **WIRE** Command Box. When the **WIRE** Command Box is set for signal (thin) wires, arcs are thin. Likewise, when **WIRE** is set for buses, arcs are fat to accommodate buses. Select the **ARC** Command Box until the desired arc orientation is displayed. Position the cursor in the Drawing Window and press and hold the mouse button or press **<Ins>**. An arc appears and can be moved throughout the Drawing Window as long as the mouse button is held pressed. The Drawing Window automatically pans if an arc is moved to any edge of the screen. Align the arc with the end point of the desired signal or bus wire and release the mouse button or press **<Ins>**. Once **BUS** is selected there is no limit to the number of arcs that can be added to a drawing. To add another arc, press and hold the mouse button (or press **<Ins>**), position the arc in the Drawing Window, and release the mouse button or press **<Ins>**.

See Also: UNDO (<ESC>)

Notes: Arcs are *cosmetic* additions to drawings and carry no signal connection information for the SCHPOST post-processors.

DRAW-COPY

Editor: Schematic

Use: To add a copy of an image already created to the current drawing.

Function: Displays the copy of the selected image in the Drawing Window.

Operation: When selected the **COPY** Command lights. Select the object to be copied with the cursor holding the left mouse button down. Drag the copied item to the desired position and place.

See Also: **EDIT-COPY** for more detailed information if needed.

Notes: **DRAW-COPY** and **EDIT-COPY** operate in the same manner.

DRAW-DOT

Editor: Schematic

Use: To identify connections of signal wires and/or objects.

Function: Symbolizes circuit connections at T-junction or 4-way signal wire intersections with solid dots. Also, breaks signal wires into two wires at the **DOT** location.

Operation: When selected, the DOT Command Box lights. Position the cursor in the Drawing Window and press and hold the mouse button or press **<Ins>**. The cursor will then disappear. The dot will be displayed where the tail of the cursor had appeared. The dot can be moved throughout the Drawing Window as long as the mouse button is held pressed. The Drawing Window automatically pans if a dot is moved to any edge of the screen. To establish a connection, position the dot over the signal wire T-intersection or 4-way junction and release the mouse button or press **<Ins>**. Once the **DOT** Command Box has been selected, there is no limit to the number of dots that can be added to a drawing. To make another connection, press and hold the mouse button (or press **<Ins>**), position the dot in the Drawing Window, and release the mouse button or press **<Ins>**.

Dots should be added after underlying connections are drawn. If a dot is placed and SCHDRAW cannot find at least one signal wire touching the dot, a beep is generated. Despite the audible warning, it is still your responsibility to remove (**UNDO**) the applicable dot.

See Also: UNDO

Notes: When a dot is placed on a signal wire, the signal wire is actually divided into two wires. This process is only noticeable when **CONFIG-WI HDR** is active.

DRAW-ERASE

Editor: Schematic

Use: To erase an image or region already created from the current drawing.

Function: Removes the selected image or region from the Drawing Window.

Operation: When selected, the ERASE Command Box lights.

To erase a single image position the cursor on the unwanted image and click the mouse button (or press <Ins> twice). The image is erased.

See Also: EDIT-ERASE, FILE-REDRAW, UNDO

Notes: Accidental erases can be reversed with UNDO (<Esc>). If two images overlap and one is erased, the other image may also appear to be erased. Use REDRAW to redraw the current drawing page and display the true remaining image.

If two or more images overlap and the wrong image is initially erased, select UNDO (<Esc>) to restore the image. Then, return to the cursor's original location and click the mouse button again (or press <Ins>). The next image is erased. If still incorrect, select UNDO (<Esc>) again and continue. SCHDRAW toggles through all images at a given location before repeating.

DRAW-GATE

Editor: Schematic

Use: To increment the gates or sections that are available for a specific component.

Function: Toggles through all gates or sections.

Operation: When selected the GATE Command Box lights. Place the cursor over the component and press the left mouse button (or press <Ins>). Continue to toggle through the different sections until you find the desired one. When no more sections are found the following message is displayed:

NO OTHER GATE/SECTION FOUND

See Also: EDIT-GATE

Notes: This command does not graphically arrange a part. The alternate gates or sections must be parts that have already been created or supplied with the SCHEMA III libraries. This command achieves the same effect as using EDIT-IMAGE to alter the DRAW-OBJECT Command line

DRAW-LABEL

Editor: Schematic

Use: To label a net.

Function: Allows labels (entered via the keyboard) to be assigned to signal wires.

Operation: When selected, the LABEL Command Box lights and the Text Window appears. Icons within the LABEL Command Box display the currently-selected label orientation. Select the LABEL Command Box until the desired label orientation is displayed. Enter the signal wire label followed by **<Enter>**. For example:

ADR00 <Enter> - Creates label **ADR00**

After the label is entered, the Text Window disappears. Position the cursor in the Drawing Window and press and hold the mouse button (or press **<Ins>**). The label appears with a small connection dot (or connection bar if **WI HDR** is active). The signal wire label can be moved throughout the Drawing Window as long as the mouse button is held pressed down. The Drawing Window automatically pans if a label is moved to any edge of the screen. Position the label over the signal wire, align the connection dot with the signal wire end point, and release the mouse button (or press **<Ins>**). The Text Window reappears and another signal wire label can be entered.

Valid LABEL text includes all letters, numbers, and punctuation characters. All letters may be entered in upper or lowercase and in medium type.

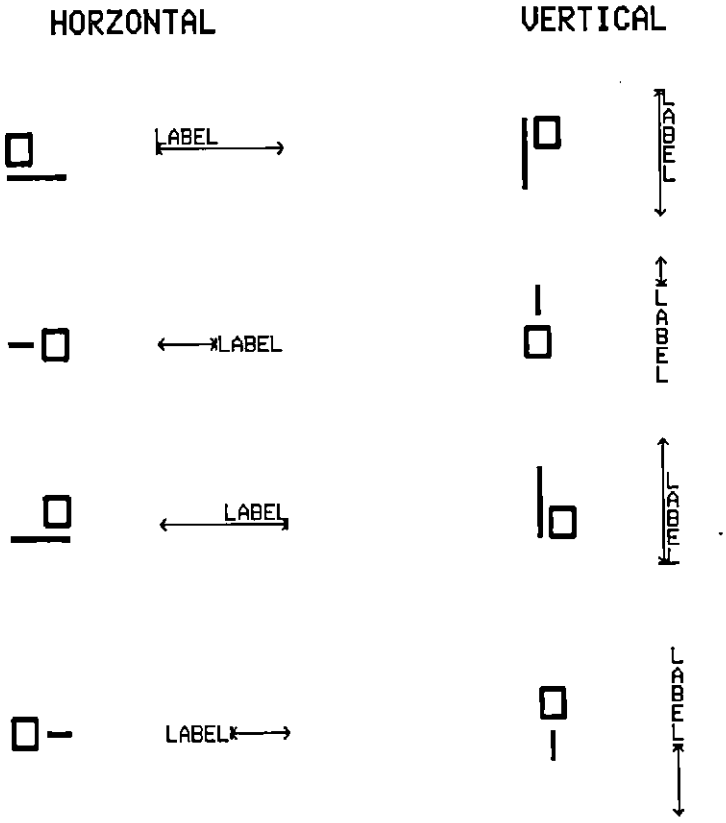
See Also: UNDO (**<ESC>**), CONFIG-WI HDR

Notes: When adding labels to signal wires, ensure label connection dots/bars touch signal wire end points so the SCHPOST post-processors can associate the labels with the signal wires.

To ensure proper SCHPOST post-processing operation, the maximum **LABEL** length should be limited to 12 characters or less.

Autoassigned labels consist of an eight digit number. Therefore, start all labels with a non-numberic character to avoid confusion.

SCHDRAW allows lower case characters in net names (labels). Some PCB products, including SCHEMA-PCB, do not accept lower case net names. To avoid any possible problems, labels can simply be entered in upper case when using SCHDRAW.


HORZONTAL
VERTICAL
LABEL
LABEL
LABEL
LABEL
LABEL
LABEL
LABEL
LABEL

DRAW-MOVE

Editor: Schematic

Use: To move an object within the schematic page.

Function: Allows any object on the current drawing to be moved.

Operation: When selected the **MOVE** Command Box lights. Place the cursor over the object to be moved. Then press and hold the left mouse button or **<Ins>**, and drag the object to its desired location on the page. Release the left mouse button or press **<Ins>** to place the object.

See Also: EDIT-MOVE

Notes: **DRAW-MOVE** operates in the same manner as **EDIT-MOVE.**

DRAW-OBJECT

Editor: Schematic

Use: To add an object from an object library to a drawing. Also, to display object library directories.

Function: Retrieves objects from object libraries or displays the selected object library directory.

Operation: When selected, the OBJECT Command Box lights and the Text Window appears.

To retrieve an object from an object library, enter the desired Parameter String in the Text Window. The Parameter String is entered in the following format:

Component Name,Reference Designator,Parameter 1,2,3,4 <Enter>

Notice that each Parameter String entry is separated by a comma character (,) with no spaces.

The **Component Name** entry is simply the component's Name Object name such as 74LS00, 8088, DDIODE, etc. The component's physical orientation and gate section can also be included after the component name, if desired. Orientation/gate section entries always follow the component name and are preceded with a backslash character (\). For example:

74LS00\B

retrieves the B-gate of a 74LS00, while:

74LS00\2C

retrieves the C-gate of the 74LS00 *DeMorgan* orientation. Object orientations are always entered numerically. If no object orientation is entered, the system defaults to orientation 1. Sub-part (gate) data is entered alphabetically (A, B, C, etc.) immediately following the orientation number. If no sub-part data is entered, the system defaults to sub-part A (gate A). For example:

74LS00 <Enter> (Defaults to Orientation 1, Gate A)
74LS00\B <Enter> (Orientation 1, Gate B)

74LS00\2 <Enter> (DeMorgan, Gate A)
74LS00\2C <Enter> (DeMorgan, Gate C)

The **Reference Designator** field follows the **Component Name** which, as mentioned above, may already include optional orientation and sub-part (gate section) fields. If the **Reference Designator** field is left empty (**<Enter>** pressed after **Component Name**, or, double commas („) entered after **Component Name**), a reference designator is automatically assigned to the component by the Schematic Editor. For example:

74LS00 <Enter> (Ref. des. auto-assigned)
74LS00,U1 <Enter> (Ref. des. U1 assigned)
DRES,,1K <Enter> (Ref. des. auto-assigned).

Do not include spaces or dashes with your reference designators. Reference designations should contain alphanumeric characters only.

The **Parameter 1,2,3,4** entries are *optional* and may or may not be included in the Parameter String, depending on the type of object used. **Parameters 1,2,3,4** are normally assigned to discrete components. Consider a sample **DRAW-OBJECT** Parameter String entry for a resistor:

DRES,R33,1.5K,10%,1/2W,100-033-000 <Enter>

The first three parameter fields (**1.5K,10%,1/2W**) are displayed with the resistor, while the last field (company part number) is not displayed but is used for the SCHPOST bill of materials output. Paragraph 3.3.4.2.3 contains additional information on these parameters.

After the Parameter String is entered in the Text Window and **<Enter>** is pressed, position the cursor in the Drawing Window. Press and hold the mouse button (or press **<Ins>**). The desired object appears and can

be moved throughout the Drawing Window as long as the mouse button is held pressed down. The Drawing Window automatically pans if an object is moved to any edge of the screen. Move the object to the desired location and release the mouse button or press **<Ins>**. The Text Window reappears and another object can be retrieved from the object library.

LIBRARY BROWSE

The DRAW-OBJECT command also allows for rapid directory searches. To display a specific object library directory, enter a directory search string in the DRAW-OBJECT Text Window. For example:

*** <Enter>** (Displays all objects in the library assigned to the drawing's configuration file LIBR1 variable)

n=* <Enter> (Displays all objects in LIBRn)

ab* <Enter> (Displays all objects that begin with **ab**, in **LIBR1**)

n=ab* <Enter> (Displays all objects that begin with **ab**, in **LIBRn**).

After **<Enter>** is pressed, the Directory Screen appears with the object names of that directory from left to right in alphanumeric order. **<Home>** moves the cursor to the first name on the screen. The **<PgUp>** key will display the previous screen and the **<PgDn>** key will display the next screen. To view a particular object, highlight that object using the mouse or arrow keys and press the left mouse button or **<Ins>** key. If the object does not fit in one screen, use the arrow keys to scroll the object. Press right mouse button or **<Esc>** to return to the previous Directory Screen and right mouse button or **<Esc>** again to return to the original drawing with the original library displayed in the Text Window.

If the object selected is to be placed in the schematic drawing, press the left mouse button or **<Enter>** key when the object is displayed. The Drawing Window will appear with the original drawing and the object name displayed in the Text Window. Press **<Enter>** and use the left mouse button or the **<Ins>** key to place the object.

See Also: EDIT-PM OBJ, BODY-PARM

Notes: Component Names may be entered in upper or lowercase, but will be automatically recognized as all uppercase. Name and Reference Designators may be entered in upper or lowercase and will be automatically converted to uppercase. P1 through P4 parameters may be entered in upper or lowercase and will appear as typed. **DRAW-GATE** and **DRAW-ROTATE** can be used if you do not wish to type in the orientation at the **DRAW-OBJECT** Command line.

DRAW-REPEAT

Editor: Schematic

Use: To step and repeat previously-issued **DRAW** commands (**WIRE, ARC, OBJECT, LABEL**).

Function: Redraws the previously drawn image at a predetermined distance from the image.

Operation: **DRAW** a wire, arc, object, or label. Select the **REPEAT** Command Box. The **REPEAT** Command Box flashes when the

previously entered **DRAW** command is repeated. If a label is repeated that contains any number of digits (example: A1, D24, DATA3, etc.), a new label is generated with an incremented value (n+1). Multiple repeats are obtained by selecting the **REPEAT** Command Box in succession.

See Also: CONFIG-RPT

Notes: REPEAT can also be activated by pressing the **<F9>** Function key.

REPEAT offset values (distances repeated images are located from original images) are set using **CONFIG-RPT**. Both X and Y repeat offset values can be assigned.

When a component is **REPEATed**, a new reference designator is automatically assigned by the Schematic Editor.

DRAW-ROTATE

Editor: Schematic

Use: To toggle through the available rotated versions of an object.

Function: Increments to the next rotated orientation.

Operation: When selected the **ROTATE** Command Box lights. Place the cursor over the component and press the left mouse button. Toggle through the different rotations until you find the desired one.

See Also: EDIT-ROTATE

Notes: The **ROTATE** command is a convenience feature that increments the DRAW-OBJECT Command line so that you don't have to use **EDIT-IMAGE** to alter the orientation of a part. Most all gated parts in SCHEMA III have rotated versions of the part included in the library.

DRAW-WIRE

Editor: Schematic

Use: To draw signal or bus wires.

Function: Symbolizes signal (thin) or bus (thick) wires.

Operation: When selected, the **WIRE** Command Box lights. Icons within the **WIRE** Command Box display the currently-selected wire type (thin or thick). Select the **WIRE** Command Box until the desired wire type is displayed. Position the cursor in the Drawing Window. Press and hold the mouse button (or press **<Ins>**) and move the cursor to draw the wire. The first cursor movement determines wire direction (X or Y). Elbow wires (L-shaped, 90-degree wires) are easily drawn. Move the cursor up and the wire moves up the Drawing Window. Move the cursor left or right and the wire routes left or right, respectively. As long as the mouse button is held pressed down, wire length and direction (in both X and Y planes) can be adjusted. If a wire is drawn in the wrong direction, reverse cursor movement to reverse the direction. The Drawing Window automatically pans if a wire is routed to any edge of the screen. When

the desired wire is displayed, release the mouse button or press <Ins>. Once WIRE is selected, there is no limit to the number of wires that can be added to a drawing. To add another wire, position the cursor in the Drawing Window, press and hold the mouse button (or press <Ins>), and move the cursor in the desired direction(s).

See Also: CONFIG-WI HDR

Notes: When UNDOing (<Esc>) wires with elbows (L-shaped, 90-degree bends), UNDO can be selected twice. Each selection undoes one section of the wire.

WIRE Command Box icon width also determines BUS width.

EDIT

Editor: Schematic/Object

Use: To edit existing graphics.

Function: In the Schematic Editor, provides access to **COPY, ERASE, GATE, IMAGE, MOVE, PM OBJ, REGION, ROTATE,** and **SELECT** Secondary Commands.

In the Object Editor, provides access to **COPY, ERASE, ID, MOVE, REGION, SELECT** and **UPDATE** Secondary Commands.

Operation: When selected, the **EDIT** Command Box lights and its Secondary Commands appear in the lower portion of the Command Window.

EDIT-COPY

Editor: Schematic/Object

Use: To add a copy of an image or a region already created to the current drawing.

Function: Displays a copy of the selected image or region in the Drawing Window.

Operation: When selected, the **COPY** Command Box lights.

To copy a single image that HAS NOT been preselected with the EDIT-SELECT command, position the cursor on the image to be copied. Press and hold the mouse button or press **<Ins>**. A copy of the image appears and can be moved throughout the Drawing Window as long as the mouse button is held pressed down. The Drawing Window automatically pans if the image is moved to any edge of the screen. Move the image copy to new location and release the mouse button (or press **<Ins>**).

To copy a single image that HAS already been selected using the EDIT-SELECT command (image slowly flashing), simply position the cursor in the Drawing Window and either press **<Ins>** or press and hold the mouse button. A copy of the image appears at the cursor. The copied image is now the selected image (image slowly flashing). Place the image as described above.

To copy a region (region must be defined), position the cursor in the Drawing Window and press and hold the mouse button (or press **<Ins>**). A copy of the region appears and can be moved throughout the Drawing Window as long as the mouse button is held pressed down. The Drawing Window automatically pans if the region is moved to any edge of the screen. Move the region copy to the new location and release the mouse button or press **<Ins>**. The REGION outline box reappears around the copied region.

See Also: EDIT-NEWREF, EDIT-REGION

Notes: When copying components that have previously assigned reference designators, the Schematic Editor automatically assigns new reference designators to the copied components.

EDIT-ERASE

Editor: Schematic/Object

Use: To erase an image or region already created from the current drawing.

Function: Removes the selected image or region from the Drawing Window.

Operation: When selected, the ERASE Command Box lights.

To erase a single image that HAS NOT been preselected with the EDIT-SELECT command, position the cursor on the unwanted image and click the mouse button (or press <Ins> twice). The image is erased.

To erase a single image that HAS been preselected with the EDIT-SELECT command (image slowly flashing), position the cursor in the Drawing Window. Next, either press <Ins> twice or click the mouse button. The image is erased and the SELECT Command Box extinguishes.

To erase a region (region must be defined), position the cursor in the Drawing Window and click the mouse button or press <Ins> twice. The predefined region is erased.

See Also: EDIT-REGION, FILE-REDRAW, UNDO

Notes: Accidental erases can be reversed with UNDO (<Esc>) only if EDIT-SELECT has not been preselected.

If two images overlap and one is erased, the other image may also appear to be erased. Use **REDRAW** to redraw the current drawing page and display the true remaining image.

If two or more images overlap and the wrong image is initially erased, select **UNDO** (**<Esc>**) to restore the image. Then, return to the cursor's original location and click the mouse button again (or press **<Ins>**). The next image is erased. If still incorrect, select **UNDO** (**<Esc>**) again and continue. SCHDRAW toggles through all images at a given location before repeating. When images are overlapped in this manner, the **EDIT-SELECT** command can be used to select the desired image.

EDIT-GATE

Editor: Schematic

Use: To increment the gates or sections that are available for a specific component.

Function: Toggles through all gates or sections.

Operation: When selected the GATE Command Box lights. Place the cursor over the component and press the left mouse button(or press **<Ins>**). Continue to toggle through the different sections until you find the desired one. When no more sections are found the following message is displayed:

NO OTHER GATE/SECTION FOUND

See Also: DRAW-GATE

Notes: This command does not graphically arrange a part. The alternate gates or sections must be parts that have already been created or supplied with SCHEMA III libraries. This command achieves the same effect as using **EDIT-IMAGE** to alter the **DRAW-OBJECT** Command line.

EDIT-ID

Editor: Object

Use: To identify Name/Body/Nested Object subcomponents.

Function: Displays the image type of the selected Name/Body/Nested Object subcomponent in the Text Window.

Operation: When selected, the ID Command Box lights. Position the cursor over the unknown object subcomponent in the Drawing Window. Press the mouse button (or press **<Ins>**) and the Text Window appears with the object's subcomponent type and identity, if applicable. Release the mouse button (or press **<Ins>**) and the Text Window disappears. If the cursor is incorrectly positioned (no object subcomponent to locate) and the mouse button or **<Ins>** is pressed, a beep is generated.

Notes: If multiple components overlap, position the cursor over the desired area and click the mouse button (or press **<Ins>**) until each overlapping component has been identified.

ID identifies the components listed in table 3-2.

Table 3-2. ID Components

ID Display	Image Type
LINE	All line types
TEXT: ABC...	Text strings
N OBJ: PINHDRx	Pin headers, where x = 1 to 6:
1 =	
2 =	
3 =	
4 =	
5 =	
6 =	
DF PI: ABC...	DF PI value
N OBJ: ABC...	Nested Object name
PARM N: ABC...	PARM N value
PARM R: ABC...	PARM R value
PARM 1-4: ABC...	PARM 1-4 value
DESC 1-4: ABC...	DESC 1-4 value
BODY: ABC...	Body Object name

EDIT-IMAGE

Editor: Schematic

Use: To edit labels, text, and DRAW-OBJECT Parameter String entries (object orientation, appearance, sub-part (gate) selection, reference designator, and parameters). This command is not displayed when EDIT-REGION is selected.

Function: For editing object orientation, appearance, sub-part selection, reference designator, and parameters. Also allows for label and text editing.

Operation: When selected, the IMAGE Command Box lights.

To edit a component that HAS NOT been preselected with the EDIT-SELECT command, position the cursor on the object to be edited. Next, either click the mouse button or press **<Ins>** twice to select the object. The Text Window appears with the current DRAW-OBJECT Parameter String. String fields can be edited with the keyboard. When finished, press **<Enter>**. The Text Window disappears and the updated object is displayed in its original location in the Drawing Window.

To edit an object that HAS been preselected using the EDIT-SELECT command (object slowly flashing), position the cursor in the Drawing Window and either press **<Ins>** or the mouse button. Edit as described above.

To edit label/text that HAS NOT been preselected with the EDIT-SELECT command, position cursor on label/text to be edited. Next, either click the mouse button or press **<Ins>** twice to select the label/text. The Text Window appears with the label/text. The label/text

can be edited with the keyboard. When finished, press **<Enter>**. The Text Window disappears and the updated label/text is displayed in its original location in the Drawing Window.

To edit label/text that HAS been preselected using the EDIT-SELECT command (label/text slowly flashing), position the cursor in the Drawing Window and either press **<Ins>** or the mouse button. Edit as described above.

See Also: DRAW-OBJECT, EDIT-SELECT

Notes: REGION is not used with IMAGE.

EDIT-LOAD

Editor: Schematic

Use: To load an existing region into the current drawing. Is only available when **EDIT-REGION** is selected. Also it can be used to load in an ASCII text file as a region. Can also be used to *MERGE* one file into a larger schematic.

Function: Loads an existing region or ASCII text file into current drawing.

Operation: **EDIT-REGION** Command Box must be lighted. Select **LOAD**. The **LOAD** Command Box lights and the Text Window appears. Enter the desired region name (less .RGN extension) followed by **<Enter>**, if trying to load a region, or type the full filename of the ASCII text file if you wish to bring in a piece of text to the screen. The

Text Window and cursor disappear. After a few seconds, the cursor reappears. Position the cursor in the Drawing Window and either press **<Ins>** or the mouse button. The cursor is replaced with the region. Position the region at the desired location and either release the mouse button or repress **<Ins>** to place the region.

Loading regions can be used to *MERGE* one schematic with a larger one. Regions are the same as a schematic. If you type the complete extension , eg. DRAWING.P01, this will load the schematic page as a region. The schematic page that is to be loaded must be at least two grid units smaller in the X and Y directions than the currently loaded page.

The browse function will operate as it does for **FILE-LOAD.** Give a ***.RGN** or ***.P01** to the command line

See Also: EDIT-NEWREF, EDIT-REGION, EDIT-SAVE, FILE-LOAD

Notes: UNDO (**<Esc>**) does not function with **EDIT-LOAD.** Reading in ASCII text files have the same memory restrictions as schematic symbols. If memory problems arise then try loading in smaller sections of the file.

EDIT-LINE

Editor: Object

Use: To edit **BODY-LINEs** in the Object Editor.

Function: Allows the user to stretch/contract a line in any direction.

Operation: When selected, the LINE command box lights. Select the end of a line use the left mouse. The line will change to the cursor color and can now be moved.

EDIT-MOVE

Editor: Schematic/Object

Use: To move images or regions within the Drawing Window.

Function: Allows any image or region on the current drawing to be moved.

Operation: When selected, the MOVE Command Box lights.

To move a single image that HAS NOT been preselected with the EDIT-SELECT command, position the cursor on the image to be moved. Press and hold the mouse button or press **<Ins>** (selected image may reposition itself slightly). The image can be moved throughout the Drawing Window as long as the mouse button is pressed. The Drawing Window automatically pans if the image is moved to any edge of the screen. Move the image to the new location and release the mouse button or press **<Ins>**. No rubberbanding occurs on single-image moves.

To move a single image that HAS been preselected with the EDIT-SELECT command (image slowly flashing), simply position the cursor in the Drawing Window and press **<Ins>** or the mouse button. The

selected image appears at the cursor location. Place as described above. The moved image remains selected (image slowly flashing).

To move a region (region must be defined), position the cursor in the Drawing Window and press and hold the mouse button (or press <Ins>). The predefined region appears and all signal wires that cross the region boundaries are temporarily erased. The region can be moved throughout the Drawing Window as long as the mouse button is held pressed down. The Drawing Window automatically pans if the region is moved to any edge of the screen. Move the region to the desired new location and release the mouse button or press <Ins>. The region outlined box reappears. All signal wires (and only signal wires) that cross region boundaries are redrawn connecting the original starting point to the new location (rubberbanding).

See Also: EDIT-REGION

Notes: When moving regions, ensure the region boundary is outside all objects. Only signal wires intersected by the region boundary are rubberbanded. If a region boundary crosses a component's pin header (stub), the wire connected to the applicable pin is considered inside the region boundary and will be rubberbanded.

EDIT-NEWREF

Editor: Schematic

Use: To maintain or change reference designators when copying or loading a region. Is only available when EDIT-REGION is selected.

Function: Maintains or changes reference designators when copying or loading a region.

Operation: EDIT-REGION Command Box must be lighted. Select NEWREF. The NEWREF Command Box toggles between ON and OFF.

If highlighted, NEWREF is ON. When using REGION-COPY or REGION-LOAD, new reference designators are generated for all of the parts contained in the region.

If not highlighted, NEWREF is OFF. When using REGION-COPY or REGION-LOAD, new reference designators are *not* generated for all of the parts contained in the region. The current reference designators remain the same.

See Also: EDIT-COPY, EDIT-LOAD

EDIT-PM OBJ

Editor: Schematic

Use: To display a component's Body Object parameters (PARMs). This command is not displayed when EDIT-REGION is selected.

Function: Displays the selected component's Body Object PARM assignments in the Text Window.

Operation: When selected, the PM OBJ Command Box lights. Position the cursor on the desired component. Press and hold the mouse button

or press **<Ins>**. The component's **PARM** values (assigned using the Object Editor **BODY-PARM** command) are displayed in the Text Window. When the mouse button is released or **<Ins>** pressed, the Text Window is removed. **PARMs** are displayed in the Text Window in the following order:

PARM N,PARM R,PARM 1 - 4

If any **PARM** is not used, the field is null (contains a comma). For example:

,REFD,VALUE,%%%%,WATT,

are the **PARMs** displayed for a resistor (**DRES**) component. The first field, the name parameter (**PARM N**), is null (a comma). The second field, the reference designator parameter (**PARM R**), is **REFD**, and so on.

Notes: **PM OBJ** is used for *viewing* only. **PARMs** are edited using the Object Editor **BODY-PARM** command.

EDIT-REGION

Editor: Schematic/Object

Use: To define a specific region of the current drawing to move, copy, erase, or save. Also used to load an existing region into the current drawing (Schematic Editor only).

Function: Identifies selected region with an outline box.

Operation: When selected, the REGION Command Box lights and the EDIT-PM OBJ and EDIT-IMAGE Command Boxes are replaced with the EDIT-SAVE and EDIT-LOAD Command Boxes, respectively.

Position the cursor to the upper left corner of the desired region. Press and hold the mouse button or press **<Ins>**. Move the cursor diagonally to the lower right corner of the desired region. As the cursor is moved, an outline box is created, enclosing the desired region. When the outline box completely encloses the desired region, release the mouse button or press **<Ins>**. The selected region is now ready to be moved, copied, erased, or saved. To save the region, refer to the EDIT-SAVE command.

See Also: EDIT-MOVE, EDIT-COPY, EDIT-ERASE, EDIT-SAVE, EDIT-LOAD

Notes: If more than 50% of an item extends beyond the REGION outline box, it will not be included in the selected region. This does not apply to wires. Any part of a wire or line extending beyond the REGION outline box is truncated when the region is moved or copied, except when rubberbanding occurs.

EDIT-ROTATE

Editor: Schematic

Use: To toggle through the available rotated versions of an object.

Function: Increments to the next rotated orientation.

Operation: When selected the ROTATE Command Box lights. Place the cursor over the component and press the left mouse buton or <Ins>. Toggle through the different rotations until you find the desired one.

See Also: DRAW-ROTATE

Notes: The ROTATE command ia a convenience feature that increments the DRAW-OBJECT Command line so that you don't have to use EDIT-IMAGE to alter the orientation of a part. Most all gated parts in SCHEMA III have rotated versions of the part included in the library.

EDIT-SAVE

Editor: Schematic

Use: To save a predefined region to disk. Only available when EDIT-REGION is selected.

Function: Saves defined region to disk.

Operation: The EDIT-REGION Command Box must be lighted and a region must be defined. Select SAVE. The SAVE Command Box lights and the Text Window appears. Enter the region name (less .RGN extension) followed by <Enter>. The Text Window disappears and the region is saved to disk as file REGION.RGN. (where REGION is the region name you enter and .RGN is the region file extension automatically appended to the region name by SCHDRAW.). The region remains defined.

A region has the same format as a schematic file. If you wish to print a section of your schematic, you could save a region off as **DRAWING.P01**, giving the full extension. Then you may use this file as you would any other schematic page.

See Also: EDIT-REGION, EDIT-LOAD

Notes: UNDO (<Esc>) does not function with **EDIT-SAVE.**

EDIT-SELECT

Editor: Schematic/Object

Use: To select a single image to move, copy, erase, or edit.

Function: Causes selected image to slowly flash.

Operation: When selected, the **SELECT** Command Box lights. Position the cursor over the desired image and either press **<Ins>** twice or click the mouse button. The selected image slowly flashes. If several images overlap at one location, the wrong image may initially be selected. If this occurs, continue to press **<Ins>** or the mouse button until the desired image is selected.

After the desired image is selected (slowly flashing) it can be **COPY**d, **MOVE**d, **ERASE**d, or edited using **EDIT-IMAGE.**

Notes: SELECT is most useful when many images occupy a small area and one image must be MOVEd, COPYd, ERASEd, or edited with EDIT-IMAGE.

Accidental erases can be reversed with UNDO (<Esc>) only if EDIT-SELECT was not preselected.

EDIT-UPDATE

Editor: Object

Use: To edit Body Object pin definitions (BODY-DF PI) and text (BODY-TEXT).

Function: Allows editing of pin definitions and text.

Operation: When selected, the UPDATE Command Box lights.

To edit Body Object DF PI/TEXT entries that HAVE NOT been preselected with the EDIT-SELECT command, position the cursor on the DF PI/TEXT entry to be edited. Next, either click the mouse button or press <Ins> twice to select the DF PI/TEXT. The Text Window appears with the current DF PI/TEXT entry. The entry can now be edited from the keyboard. When finished, press <Enter>. The Text Window disappears and the updated DF PI/TEXT is displayed in its original location in the Drawing Window.

To edit Body Object DF PI/TEXT entries that HAVE been preselected with the EDIT-SELECT command (DF PI/TEXT slowly flashing), position the cursor in the Drawing Window and either press <Ins> or the mouse button. Edit as described above.

See Also: EDIT-SELECT

Notes: The UNDO (<Esc>) command does not function with any EDIT-UPDATE operation.

EDIT-WIRE

Editor: Schematic

Use: To alter and modify thin signal wires, DRAW-WIRE or thin NOTES-LINEs.

Function: Allows the cleanup of wire after a REGION-MOVE and the adjustment of lines.

Operation: When selected, the WIRE command box lights. NOTES-LINEs can be expanded or shrunk by selecting and dragging an end-point. Thin signal wires can be expanded by selecting and dragging an end-point. If the two wires terminate at the end-point then both are modified. However, if a single wire end-point is grabbed, two wires can be edited in the usual manner. By working from the two end points of a wire the user can modify the wire to suit their needs.

See Also: EDIT-LINE in the Object Editor

FILE

Editor: Schematic

Use: To load, save, redraw, merge, or print a drawing. Also, to exit SCHDRAW.

Function: Provides access to **ED OBJ, LOAD, POP, PRINT, PUSH, QUIT, REDRAW,** and **SAVE** Secondary Commands.

Operation: When selected, the **FILE** Command Box lights and **ED OBJ, LOAD, PRINT, QUIT, REDRAW,** and **SAVE** Secondary Commands appear in the lower portion of the Command Window.

FILE-ED OBJ

Editor: Schematic

Use: To invoke the Object Editor.

Function: Provides access to the Object Editor and associated commands.

Operation: When selected, a decision verification prompt appears in the Error Window. Press **<Enter>** to invoke the Object Editor or select any command to cancel the **ED OBJ** request. When the Object Editor is invoked, the following Object Editor Primary Commands appear in the Command Window:

BODY
CONFIG
EDIT
HELP
LIBR
NAME
ZOOM

Notes: The cursor briefly disappears while the Object Editor is invoked.

FILE-LOAD

Editor: Schematic

Use: To load a drawing page into the Schematic Editor. Also, to view DOS directories.

Function: Displays the selected drawing in the Drawing Window or displays a list of DOS files.

Operation: When selected, the **LOAD** Command Box lights and the Text Window appears with the previously entered **LOAD** or **SAVE** text string (**TEMP.P01** is displayed if nothing has been previously entered).

To load a drawing page into the Schematic Editor, enter the desired drawing file name with path, if needed. If no file extension is entered, a default extension (**.P01**) is used. Press **<Enter>** to load the drawing file. The Text Window and cursor disappear. If the drawing file has an associated configuration file (DRAWING.CFG file), the drawing's

configuration file data is loaded and used by SCHDRAW. If a DRAWING.CFG file does not exist, the SCHEMA.CFG configuration file data is loaded and used by SCHDRAW. After the drawing has been loaded, the LOAD Command Box extinguishes and the upper left corner of the drawing is displayed in the Drawing Window.

FILES BROWSE

To list for selection all the drawings available, enter:

***.P* or *.* <Enter>**

Both valid DOS wildcard characters (? and *) are supported (refer to your DOS manual).

The Text Window and cursor disappear and the specified drawing files are displayed in the Drawing Window. Use the mouse or <Arrow keys to highlight the desired file to load and press the left mouse or <Enter>. For example, to display the names of all files in the current directory enter:

***.* <Enter>**

See Also: FILE-PUSH and FILE-POP

Notes: Using **LOAD** also resets the **PUSH/POP** stack to an empty status prior to loading a new drawing.

FILE-POP NN

Editor: Schematic

Use: To return to and load a drawing file from a hierarchical level above the current drawing file.

Function: Jumps to next hightest level of hierarchy.

Operation: Hiearchical drawing file names are placed on the stack when FILE-PUSH command is used to descend into a lower level of hierarchy in the schematic. The NN represents the level of hierarchy of your current drawing page. When selected the POP NN Command Box lights and text window displays the drawing filename for that level. As you toggle through the stack the level number and the drawing files on the stack decrement.

See Also: FILE-PUSH

Notes: The level number is relative to the last drawing file loaded with FILE-LOAD since FILE-LOAD resets the file name stack to empty.

FILE-PRINT

Editor: Schematic

Use: To print the current drawing page in either foreground or background (hard disk systems only). The drawing is printed at the same zoom level as currently displayed on the screen.

Function: Creates a hardcopy of the selected drawing.

Operation: When selected, the **PRINT** Command Box lights and the Text Window displays the print command line:

PRN

If enter is pressed then the output will be sent to the port, **PRN.**

If the **PRINT** command is pressed again, then the filename with the extension, **TXX** appears.

DESTINATION <Space> /OPTION

where **DESTINATION** is the destination of the print operation and **/OPTION** is print options.

The destination is either a file name (spool drawing to a file and initiate background printing) or the letters **PRN** which initiates foreground printing. One print command line destination is the current filename with a .Txx file extension. .Txx print files are lengthy, i.e., up to 600K for a D-size drawing (A mode, Epson printer). Thus, background printing can only be done from *hard disk systems*.

The DOS PRINT utility **MUST** be initialized prior to invoking SCHDRAW to ensure proper background printing. Refer to your DOS PRINT documentation for specific information on the DOS PRINT utility. SCHDRAW invokes the standard DOS PRINT utility which operates as defined in DOS documentation.

Print options are specified by following the destination with a **<Space>**, slash character (/), and the desired print **OPTION.**

OPTION defines the print "mode" and orientation of the output.

Each printer driver supports one or more modes of output. If no mode is specified, or, if an invalid mode is selected, the printer drivers default to mode **A**. In most cases, mode **A** generates a drawing of approximately 3/8 template scale while mode **D** generates a drawing of 3/4 template scale. Individual drivers may support other modes which provide different scaling and/or resolutions. For example, the Epson driver generates the same scale drawing using both modes **C** and **D**; however, mode **C** prints in about half the time as mode **D**, but with much less resolution. Refer to Appendix E, *Hardware Requirements*, to determine what modes are supported by the printer driver you have selected.

All printer drivers are capable of printing a drawing in one of two orientations. The default orientation generates a "sideways" printout, where the x-direction of the drawing is printed on the y-direction of the paper. Specifying the **R** option generates a portrait printout, where the x-direction of the drawing is printed on the x-direction of the paper.

Some sample **PRINT** command lines are:

MEMORY.T02 <Enter>
(Creates a temporary print file named **MEMORY.T02** and prints the file in the normal mode, in background.)

PRN<Space>/D <Enter>
(Produces a full-size print in foreground. No print file is created.)

MEMORY.T02 <Space>/CR <Enter>
(Creates a temporary print file named **MEMORY.T02**, prints the file in **C** mode, portrait style, in background.)

MEMORY.T02 <Space>/R <Enter>
(Creates a temporary print file named **MEMORY.T02** and prints the

file in a portrait format, A mode, in background.)

PRN<Space>/rc <Enter>
(Produces full-size print file in portrait format,
C mode, in foreground.)

While a temporary print file is being created for background printing or printing is occurring in foreground, **<Esc>** can be pressed to abort the operation. An abort prompt is displayed in the Text Window when **<Esc>** is pressed.

Output can be directed to any **COM** or **LPT** port by typing to the command line for example, instead of **PRN** type **LPT2** or **COM2**

See Also: Chapter 4, *SCHPLOT/SCHPRINT.*

Notes: All **FILE-PRINT** operations print drawings as they are displayed. Thus, if **V GRID** and/or **WI HDR** are enabled, they are printed as part of the drawing.

FILE-PUSH

Editor: Schematic

Use: To descend into a lower level of hierarchy, or to load non-electrical schematics or document files.

Function: Invoke lower levels of hierarchy, or load non-electrical schematics or text files.

Operation: When selected the **PUSH** Command Box lights. Place the cursor over the hierarchical object on the current drawing page. Select the object with the left mouse button or **<Ins>**. The page corresponding to the hierarchical object selected, appears in the Text window. Press **<Enter>** to load the new drawing page name, and "push" the filename onto the stack.

There are two ways that a drawing can be pushed to a lower level.

One method is if the selected part is a hierarchical object. If this object is defined as per the guidelines of creation, then the hierarchical schematic filename that the object represents appears in the Text Window waiting for **<Enter>** to be pressed.

The other method is to place the filename of a schematic or a text file in PARM 4 of an object. If the file is a schematic then it is loaded. If the file is an ASCII text file then the software automatically senses that it is text and will create a page that can accomodate it. This method is useful if you wish to have the schematic carry additional documentation, or display for example the "glue logic" of a PAL in equation or schematic form. The text or schematics are not used in the generation of any Post-Processing. The SCHEMA libraries contain a document icon, called DOCUMENT. This part can be used as follows:

DOCUMENT,,Manufacturing,Notes,,MANUFAC.TXT

See Also: FILE-POP NN

Notes: SCHEMA creates a page to accomodate a text file. However, be aware that huge text files may cause SCHEMA to attempt to create a file that is too large to load into memory.

If the file you push down into, has a configuration file then that config file will be loaded. If at a later time you push down to a file that has no config file then the original config that was loaded upon the FILE-LOAD will be used.

FILE-QUIT

Editor: Schematic

Use: To exit SCHDRAW and return to the Manager or DOS.

Function: Terminates SCHDRAW.

Operation: When selected, the QUIT Command Box lights and a prompt appears in the Text Window. Press <Enter> to terminate SCHDRAW or select any command box to cancel the QUIT request.

Notes: SAVE your drawing before using QUIT or your work will be lost when SCHDRAW is terminated.

FILE-REDRAW

Editor: Schematic

Use: To redraw the current drawing.

Function: Redraws the current drawing.

Operation: When selected, the **REDRAW** Command Box lights and the cursor disappears while the drawing is redrawn. After the drawing is redrawn, the cursor reappears and the **REDRAW** Command Box extinguishes.

Notes: **REDRAW** is generally used after multiple images have shared the same screen location and one of the images has been erased, leaving a distorted image. **REDRAW** restores distorted images to their original appearance. After **REDRAW** is issued, the true drawing contents are displayed. Thus, **REDRAW** works in a manner similar to saving a file to disk and then reloading the file.

FILE-SAVE

Editor: Schematic

Use: To save the current drawing to disk.

Function: Outputs the current drawing to the applicable storage device.

Operation: When selected, the **SAVE** Command Box lights and the Text Window appears with the previously-entered **LOAD** or **SAVE** text string (**TEMP.P01** is displayed if nothing has previously been entered).

To save a drawing page to disk, enter the desired drawing file name with path, if applicable. Be sure to include the appropriate page number extension (.Pxx). If no page number extension is specified, SCHDRAW saves the drawing with a **.P01** extension. Press **<Enter>** to save the

drawing file. The Text Window and cursor disappear. After the drawing has been saved, the SAVE Command Box extinguishes.

Notes: If the saved file already exists, the Schematic Editor automatically creates a back-up file by changing the existing file name extension to .Bxx (if original file extension was .Pxx) and saves the new file with the specified page number extension.

If a drawing has auto-reference designators, they are updated to reflect the new page number of the saved file.

Characters may be in upper or lowercase when entering the file name and path, but will be automatically converted to uppercase.

HELP

Editor: Schematic/Object

Use: To identify the keyboard keys recognized by SCHDRAW, and gives a screen of HELP information for each SCHDRAW command.

Function: Displays Help information for SCHDRAW.

Operation: When selected, the HELP Command Box lights and remains lighted. A list of keyboard keys recognized by SCHDRAW is displayed in the Drawing Window. Next a Primary Command can be selected and a Help screen for that Primary will be displayed. Then if a Secondary Command is selected a HELP screen for that Primary-Secondary combination is displayed. The HELP mode can be exited by selecting HELP again.

LIBR

Editor: Object

Use: To load, save, or redraw an object. Also used to clear the Drawing Window and quit SCHDRAW.

Function: Provides access to **ED SCH, LOAD, NEW, QUIT, REDRAW,** and **SAVE** Secondary commands.

Operation: When selected, the **LIBR** Command Box lights and its associated Secondary Commands appear in the lower portion of the Command Window.

LIBR-ED SCH

Editor: Object

Use: To invoke the Schematic Editor.

Function: Provides access to the Schematic Editor and associated commands.

Operation: When selected, a decision verification prompt appears in the Text Window. Press **<Enter>** to invoke the Schematic Editor or select any command to cancel the **ED SCH** request. When the Schematic Editor is invoked, the following Schematic Editor Primary Commands appear in the Command Window:

CONFIG
DRAW
EDIT
FILE
HELP
NOTES
SEARCH
ZOOM

Notes: When the Object Editor is exited for the Schematic Editor, the Drawing Window is cleared. Thus, be sure to **SAVE** the current object before exiting the Object Editor or your work will be lost. Upon returning to the Schematic Editor, the upper left corner of the current drawing appears in the Drawing Window.

LIBR-LOAD

Editor: Object

Use: To load a Name, Body, or Nested Object into the Object Editor from the desired object library for editing or viewing.

Function: Retrieves an object from desired object library and displays the object in the Drawing Window.

Operation: When selected, the **LOAD** Command Box lights and the Text Window appears. If an object has been previously loaded or saved, the library number and object name are displayed in the Text Window. If no load or save operation has occurred, the Text Window is empty. Enter

the desired object name followed by **<Enter>**. Optionally, the desired library can be specified by preceding the object name with:

n=

where **n** is the corresponding configuration file **LIBR** library variable number (**1 - 9**). If a library is not specified, SCHDRAW searches the object libraries for the desired object in numerical order. The library assigned to the drawing's configuration file Library 1 variable (**LIBR1**) is searched first, followed by the library assigned to **LIBR2**, and so on until the object is located. When the object is located, the Text Window disappears and the desired object appears in the Drawing Window.

Some sample **LOAD** entries are:

74LS00 <Enter> - Search all libraries starting with **LIBR1** for 74LS00 object

3=74LS00 <Enter> - Search only **LIBR3** for 74LS00 object.

5=* <Enter> perrforms a library **Browse** for **LIBR5**
Library objects can also be viewed through **LOAD.** Refer to the Schematic Editor **DRAW-OBJECT** command for additional information on parts browsing.

See Also: DRAW-OBJECT

Notes: When **LOAD** is executed, the existing object (entire contents of Drawing Window) in the Object Editor is cleared and the new object is loaded. Save the current object, if desired, before executing **LOAD.** The existing object is also cleared if **LOAD** is used to view library objects.

Characters may be in upper or lowercase when entering the file name and path, but will be automatically converted to uppercase.

LIBR-NEW

Editor: Object

Use: To clear the Object Editor Drawing Window contents.

Function: Erases the Drawing Window.

Operation: When selected, the NEW Command Box lights and the Drawing Window is cleared. The NEW Command Box automatically extinguishes after the Drawing Window is cleared.

See Also: LIBR-SAVE

Notes: All objects in the Drawing Window are cleared when NEW is executed. So be sure to SAVE the current object, if desired, before issuing NEW.

UNDO does not function with NEW.

LIBR-QUIT

Editor: Object

Use: To exit SCHDRAW and return to the Manager or DOS.

Function: Exits the Object Editor and returns to the Manager or DOS.

Operation: This command operates identically to the Schematic Editor **FILE-QUIT** command.

See Also: **FILE-QUIT**

Notes: **SAVE** the current object and Schematic Editor drawing before **QUIT**ing or your work will be lost.

LIBR-REDRAW

Editor: Object

Use: To redraw the Drawing Window.

Function: Redraws the Drawing Window.

Operation: This command operates identically to the Schematic Editor **FILE-REDRAW** command.

See Also: **FILE-REDRAW**

LIBR-SAVE

Editor: Object

Use: To save the current Name, Body, or Nested Object to the desired object library under an assigned name.

Function: Saves the current object to disk.

Operation: When selected, the SAVE Command Box lights and the Text Window appears. If an object has been previously loaded or saved the library number and object name are displayed in the Text Window. If no load or save operation has occurred, the Text Window is empty. Enter desired object name followed by **<Enter>**. Optionally, the desired library to save to can be specified by preceding the object name with:

n=

where **n** is the corresponding configuration file **LIBR** library variable number (**1 - 9**). If a library is not specified, SCHDRAW saves the object to the library assigned to the configuration file **LIBR1** variable.

Some sample **SAVE** entries are:

74LS00 <Enter> - Saves the current object as 74LS00 to the library assigned to the **LIBR1** configuration file variable.

3=74LS00 <Enter> - Saves the current object as 74LS00 to the library assigned to the **LIBR3** configuration file variable.

See Also: DRAW-OBJECT

Notes: When SCHDRAW is initialized, the SCHEMA III object libraries are opened in a read-only mode. **SAVE** opens a temporary version of the selected read-only library for write use only. On completion of **SAVE**, the temporary version is closed.

Characters may be in upper or lowercase when entering the file name and path, but will be automatically converted to uppercase.

NAME

Editor: Object

Use: To create/edit Name Objects.

Function: When selected, provides access to the **BODY** and **DESC** Secondary Commands.

Operation: When selected, the **NAME** Command Box lights and the **BODY** and **DESC** Secondary Commands appear in the lower portion of the Command Window.

NAME-BODY

Editor: Object

Use: To assign one or more Body Objects to a Name Object.

Function: Displays the Name Object's associated Body Object prefix in the Drawing Window.

Operation: When selected, the **BODY** Command Box lights and the Text Window appears with the previously-entered **BODY** field (if applicable). Edit/enter the desired **BODY** field followed by **<Enter>**. The **BODY** field is the prefix of the actual Body Object name. Do not include the Body Object's suffix (**-na**) with the **BODY** field. Position the cursor in Drawing Window and press and hold the mouse button or press **<Ins>**. The **BODY** field appears and can be moved throughout the Drawing

Window as long as the mouse button is pressed. The Drawing Window automatically pans if the Body Object prefix is moved to any edge of the screen. Release the mouse button or press <Ins> when the **BODY** field is in the desired position.

Notes: The actual, physical location of the **BODY** field in the Drawing Window is unimportant.

A good example of **BODY** field use is illustrated in the 74LS00 Name Object. This Name Object uses T00 as its **BODY** field. T00 allows the T00-1A, T00-1B, T00-1C, T00-1D, T00-2A, T00-2B, T00-2C, and T00-2D Body Objects to be associated with the 74LS00 component.

NAME-DESC

Editor: Object

Use: To define a component's part number and other descriptions for the SCHPOST post-processors.

Function: Displays Name Object **DESC** fields in the Drawing Window.

Operation: When selected, the **DESC** Command Box lights and the Text Window appears with the current **DESC** field. An icon within the **DESC** Command Box indicates the currently-selected description type: **1, 2, 3,** or **4**. Select the **DESC** Command Box until the desired description type is displayed. As the command box is selected, the currently-assigned **DESC** field is displayed in the Text Window. To add a new description or modify a current field, enter data in the Text Window followed by <Enter>. The Text Window disappears. Position the cursor in the

Drawing Window and press and hold the mouse button (or press **<Ins>**). The new **DESC** field appears and can be moved throughout the Drawing Window as long as the mouse button is pressed. The Drawing Window automatically pans if a **DESC** field is moved to any edge of the screen. Move description to desired location and release the mouse button or press **<Ins>**.

Notes: Omation has predefined the **DESC 1** field as the in-house (company) part number and the **DESC 2** field as the component's description (i.e., **74LS00 - LS TTL QUAD TWO-INPUT NAND**). The **DESC 3** and **DESC 4** fields are shipped undefined.

DESC lines can be replaced by **DRAW-OBJECT** Parameter String entries by entering **<Pn>** (where **P** stands for Parameter and **n** is the Parameter number, 1 to 4) as **DESC** text. For example, if a resistor's Schematic Editor **DRAW-OBJECT** Parameter String is:

DRES,R1,1K,10%,1/2W

and the resistor's Name Object **DESC 2** entry is:

RESISTOR,<P1>,<P2>,<P3>

the SCHPOST bill of materials **D2** field output is printed as:

RESISTOR,1K,10%,1/2W

The actual, physical location of **DESC** text in the Drawing Window is not important or critical.

DESC entries are not displayed with components in the Schematic Editor.

DESC entries may be in upper or lowercase.

NOTES

Editor: Schematic

Use: To add non-schematic graphics such as timing relationships, textual comments, title block information, etc. to drawings.

Function: Provides access to **TEXT, LINE, CIRCLE, MOVE, COPY,** and **ERASE** Secondary Commands.

Operation: When selected, the **NOTES** Command Box lights and the **TEXT** and **LINE** Secondary Commands appear in the lower portion of the Command Window.

Notes: **NOTES** operations are ignored by the SCHPOST post-processors.

NOTES-COPY

Editor: Schematic

Use: To add a copy of labels/text, an image, or a region already created to the current drawing.

Function: Displays a copy of the selected image or region in the Drawing Window.

Operation: When selected, the **COPY** Command Box lights.

Position the cursor on the image to be copied. Press and hold the mouse button or press **<Ins>**. A copy of the image appears and can be moved throughout the Drawing Window as long as the mouse button is held pressed down. The Drawing Window automatically pans if the image is moved to any edge of the screen. Move the image copy to new location and release the mouse button (or press **<Ins>**).

See Also: EDIT-COPY

Notes: When copying components that have previously assigned reference designators, the Schematic Editor automatically assigns new reference designators to the copied components.

NOTES-ERASE

Editor: Schematic/Object

Use: To erase labels/text, an image, or region already created from the current drawing.

Function: Removes the selected image or region from the Drawing Window.

Operation: When selected, the ERASE Command Box lights.

Position the cursor on the unwanted image and click the mouse button (or press **<Ins>** twice). The image is erased.

See Also: EDIT-ERASE, UNDO

Notes: Accidental erases can be reversed with UNDO (<Esc>).

If two images overlap and one is erased, the other image may also appear to be erased. Use REDRAW to redraw the current drawing page and display the true remaining image.

If two or more images overlap and the wrong image is initially erased, select UNDO (<Esc>) to restore the image. Then, return to the cursor's original location and click the mouse button again (or press <Ins>). The next image is erased. If still incorrect, select UNDO (<Esc>) again and continue. SCHDRAW toggles through all images at a given location before repeating.

NOTES-CIRCLE

Editor: Schematic

Use: To add thin arcs or circles to drawings.

Function: Symbolizes thin arcs or circles.

Opeation: This command operates identically to the Object Editor BODY-CIRCLE command.

See Also: BODY-CIRCLE

Notes: The SCHPOST post-processors do not interpret arcs or circles.

NOTES-LINE

Editor: Schematic

Use: To add orthogonal thin, thick, dotted, and dashed lines to drawings for notational purposes. When **CONFIG-DIAG** is selected, allows thin diagonal lines to be drawn.

Function: Symbolizes thin, thick, dotted, and dashed lines.

Operation: This command operates identically to the Object Editor BODY-LINE command.

See Also: BODY-LINE

Notes: The SCHPOST post-processors do not interpret lines. Use NOTES-LINE for applications such as notes, block diagrams, special symbols, etc.

NOTES-MOVE

Editor: Schematic/Object

Use: To move labels/text, images, or regions within the Drawing Window.

Function: Allows any image or region on the current drawing to be moved.

Operation: When selected, the **MOVE** Command Box lights.

Position the cursor on the image to be moved. Press and hold the mouse button or press **<Ins>** (selected image may reposition itself slightly). The image can be moved throughout the Drawing Window as long as the mouse button is pressed. The Drawing Window automatically pans if the image is moved to any edge of the screen. Move the image to the new location and release the mouse button or press **<Ins>**. No rubberbanding occurs on single-image moves.

See Also: EDIT-MOVE, DRAW-MOVE

NOTES-TEXT

Editor: Schematic

Use: To add text (not associated with **LABEL**s) to drawings. Also used to bring configuration file title block variable **F1 - F8** assignments into the current drawing.

Function: Allows upper or lowercase; small, medium or large text in vertical and horizontal orientations to be added to drawings. Greek and International characters are also included.

Operation: When selected, the **TEXT** Command Box lights and the Text Window appears. Icons within the **TEXT** Command Box display the currently-selected text size (small, medium or large). Select the **TEXT** Command Box until the desired text size is displayed. Enter the desired

text in the Text Window followed by **<Enter>**. Position the cursor in the Drawing Window and press and hold the mouse button or press **<Ins>**. Text appears in the Drawing Window and can be moved throughout the Drawing Window as long as the mouse button is pressed. The Drawing Window automatically pans if text is moved to any edge of the screen. Move the text to the desired location and release the mouse button or press **<Ins>**.

Text can be aligned vertically, if desired. To align text vertically, enter a backslash character (\) followed by **<Space>** and applicable text. The amount of space between vertical text characters can also be varied. To vary space between vertical **TEXT** characters, enter a backslash followed by a numeric value (maximum of 32), **<Space>**, and applicable text. The numeric value corresponds to the total number of pixels between the top of one character and the top of the next character immediately below it. Each medium text letter is formed from nine vertical pixels. If standard spacing is desired, just enter a backslash, **<Space>**, and desired text. SCHDRAW defaults character spacing to 11 pixels.

Add Title Block Variables

To manually place configuration file title block variable (**F1 - F8**) assignments in the drawing title block, enter:

<Fn>....<Enter> (Do not use your Function keys; literally enter < followed by F, a number (**n**), and >....)

where: **n = 1 to 8.** Enough periods must be entered after **<Fn>** to match the length of the actual variable assigned to the associated configuration file title block variable. After **<Enter>** is pressed, position the cursor in the Drawing Window and press and hold the mouse button (or press **<Ins>**). The applicable title block variable

originally assigned in the drawing's configuration file appears and can be placed in its appropriate title block area. If no such variable text exists, three question mark characters (???) appear when the mouse button or **<Ins>** is pressed.

There are three extra title block variables that are available but are automatically updated and do not use the values F1-F8. These are :

<FT> - Time/ Date Stamp
<FF> - File name and path
<FP> - File page number eg. 2 of 6

These can be placed on the title block in the same manner as the other variables. SCHPLOT and SCHPRINT will automatically fill in these variables when output.

EXAMPLES

Some **TEXT** entry examples are as follows:

XYZ <Enter> - **XYZ** is aligned horizontally

\<Space>XYZ <Enter> - **XYZ** is aligned vertically

\14<Space>XYZ <Enter> - **XYZ** is aligned vertically, with 14 pixels inserted between characters

<F2>....<Enter> - Configuration file **F2** title block variable is used as text.

See Also: EDIT-IMAGE

Notes: NOTES-TEXT entries can be edited using the EDIT-IMAGE command.

Spacing between horizontal text is fixed and can not be varied. Once a configuration file title block variable (F1 - F8) has been placed in a drawing, it cannot be distinguished from other TEXT.

SEARCH

Editor: Schematic/Object

Use: To locate a particular coordinate, object, reference designator, label or text in a drawing and allow the data to be edited.

Function: Provides access to **X,Y, NAME, REFD, LABEL** and **TEXT** Secondary Commands.

Operation: Select **SEARCH.** The Command Box lights and several Secondary Commands appear in the lower portion of the Command Window.

Notes: When the item is found, the cursor and the item will be centered automatically in the Drawing Window.

SEARCH-LABEL

Editor: Schematic/Object

Use: To locate a particular label in a drawing.

Function: Quickly finds a specified label or group of labels in a drawing and allows the label(s) to be edited.

Operation: Select LABEL. The Command Box lights and the Text Window appears. Enter the name of the label and press **<Enter>**. For example, to find the label, DATA, enter:

DATA <Enter>

The cursor appears on the label and it is centered in the Drawing Window. The Text Window appears with the label. The label can be edited with the keyboard. When finished, press **<Enter>**. The Text Window disappears and the updated label is displayed in its original location in the Drawing Window. To find the next DATA, press the mouse button or the **<Ins>** key twice. Note that you can search for the next occurrence without performing an edit by simply pressing the mouse button (i.e., don't press **<Enter>**).

See Also: DRAW-LABEL, EDIT-IMAGE

Notes: LABEL is upper and lower case sensitive. SEARCH only accepts 20 characters. If more than 20 are needed, enter 19 characters followed by an *. Both horizontal and vertical labels are found.

SEARCH-NAME

Editor: Schematic/Object

Use: To locate a particular object in a drawing.

Function: Quickly finds a specified object or group of objects in a drawing and allows the DRAW-OBJECT parameter string to be edited.

Operation: Select **NAME**. The Command Box lights and the Text Window appears. Enter the object name and press **<Enter>**. For example, enter:

74LS00 <Enter>

The cursor appears on the object and it is centered in the Drawing Window. The Text Window appears with the current **DRAW-OBJECT** parameter string. String fields can be edited with the keyboard. When finished, press **<Enter>**. The Text Window disappears and the updated object is displayed in its original location in the Drawing Window. To find the next 74LS00, press the mouse button or press **<Ins>** twice. Note that you can search for the next occurrence of an object without performing an edit by simply pressing the mouse button (i.e., don't press **<Enter>**).

To find all objects beginning with 74LS, then enter:

74LS* <Enter>

The cursor appears on the first 74LS object. Press the left mouse button or the **<Ins>** key and the cursor appears on the next 74LS object. Repeat this until all 74LS objects are found.

The wildcard, ?, can be used, but a character must reside in that digit for a match to be found. For example, enter:

74LS7?

All 74LS7 objects (74LS70 - 74LS79) that are in the drawing can be found when pressing the mouse button or the **<Ins>** key.

See Also: **DRAW-OBJECT, EDIT-IMAGE,** Chapter 3, *SCHDRAW*

Notes: NAME searches for all orientations and gate sections of the specified part.

SEARCH-REFD

Editor: Schematic/Object

Use: To locate a particular reference designator in a drawing.

Function: Quickly finds a specified reference designator or group of reference designators in a drawing and allows the **DRAW-OBJECT** parameter string to be edited.

Operation: Select **REFD**. The Command Box lights and the Text Window appears. Enter the desired reference designator and press **<Enter>**. For example, to find R1, enter:

R1 <Enter>

The cursor appears on the object and it is centered in the Drawing Window. The Text Window appears with the current **DRAW-OBJECT** parameter string. String fields can be edited with the keyboard. When finished, press **<Enter>**. The Text Window disappears and the updated object is displayed in its original location in the Drawing Window. To find the next occurrence of R1, press the mouse button or press **<Ins>** twice. Note that you can search for the next occurrence without performing an edit by simply pressing the mouse button (i.e., don't press **<Enter>**).

To find all objects with auto-assigned reference designators, enter:

.* <Enter>

See Also: DRAW-OBJECT, EDIT-IMAGE

SEARCH-TEXT

Editor: Schematic/Object

Use: To locate particular text in a drawing.

Function: Quickly finds the specified text in a drawing and allows the text to be edited.

Operation: Select **TEXT**. The Command Box lights and the Text Window appears. Enter the text to be found and press **<Enter>**.

The cursor appears on the text and it is centered in the Drawing Window. The Text Window appears with the text. The text can be edited with the keyboard. When finished, press **<Enter>**. The Text Window disappears and the updated label is displayed in its original location in the Drawing Window. To find the next text string, press the mouse button or the **<Ins>** key twice. Note that you can search for the next occurrence without performing an edit by simply pressing the mouse button (i.e., don't press **<Enter>**).

See Also: NOTES-TEXT, EDIT-IMAGE

Notes: TEXT is upper and lower case sensitive. SEARCH will only accept 20 characters. If more than 20 are needed, enter the first 19 characters followed by an *. Both horizontal and vertical text is found and all font sizes are found.

SEARCH-X,Y

Editor: Schematic/Object

Use: To locate a particular coordinate in a drawing.

Function: Quickly finds a particular coordinate.

Operation: Select the X,Y Secondary Command. The X,Y Command Box lights and the Text Window appears. Enter the desired coordinate and press **<Enter>**. For example, enter:

13.05,15.10

The cursor and the location being searched will be centered automatically in the Drawing Window.

Notes: All decimal value coordinates must be between .00 and .15 and must have a digit entered before the decimal point (e.g., 0.13). The decimal portion of the coordinate is optional.

SEARCH-X,Y is useful to find the X,Y coordinates listed in SCHPOST error messages, etc.

UNDO

Editor: Schematic/Object (non-menu)

Use: To undo the last command executed.

Function: Reverses most SCHDRAW commands.

Operation: After issuing an incorrect command, immediately click the right mouse button, to execute the UNDO sequence, or press <Esc>. For example, if the previously entered command added text or an object to the Drawing Window, UNDO erases the text or object. If the command was an erase-type operation, UNDO restores the erased graphics to the Drawing Window. If an object was moved within the Drawing Window, UNDO moves the object back to its original location. UNDO does not function with LIBR, FILE, EDIT-IMAGE, REGION-LOAD or if EDIT-SELECT has been preselected.

Notes: Normally, UNDO can only be issued once. However, you can undo an UNDO operation. An exception is undoing elbow (L-shaped) wires or lines. UNDO can be issued twice (once for each wire/line segment) when undoing elbow wires or lines. If UNDO is issued erroneously, a beep is generated. The right mouse button can be clicked to execute UNDO. Press UNDO or <Esc> again and UNDO will undo itself.

When undoing regions, the entire page is redrawn. Thus, region UNDOs may take several seconds. A region UNDO restores the drawing to the state it was in when the region was *first defined.* Thus, if several operations (move, copy, erase, etc.) are done on a single region and the most recent operation is undone, all previous operations are also undone.

ZOOM

Editor: Schematic/Object

Use: To view and edit drawings from four different perspectives.

Function: Provides four different views of the current drawing.

Operation: Select the **ZOOM** Command Box to adjust the Drawing Window viewing area to the position with which you feel most comfortable. Four perspectives are available, **1:1, 2:1, 4:1, 8:1, VGA.** *Drawing and editing can occur from any perspective.*

When **ZOOM** and a perspective other than 1:1 are selected, both boxes light and a much larger, less detailed portion of the drawing is displayed with a transparent, outlined box in the center of the Drawing Window. Position the cursor to the upper left corner of the area you would like to view and press the mouse button or **<Ins>**. The upper left corner of the outlined box snaps to the cursor position. The outlined box can be moved throughout the Drawing Window *as long as the mouse button is held pressed down.* The Drawing Window automatically pans if the outlined box is moved to any edge of the screen. Release the mouse button or press **<Ins>** when the outlined box encloses the desired area. Next, select **ZOOM** and the entire Drawing Window contains a detailed view of the drawing area. VGA is useful for users with hi-resolution graphics that may want more detail when creating an object or placing lines or text.

Notes: There are two "Hot" keys for **ZOOM**. **<F7>** Zooms in one level and **<F8>** Zooms out one level.

When **ZOOM** and 2:1 are active, the area displayed is two times larger along the horizontal and vertical planes of the Drawing Window than the area displayed when **ZOOM** is inactive. The area displayed is four times larger in each direction when 4:1 is active and eight times when 8:1 is active.

The amount of resolution provided by **ZOOM** is determined by your graphics adapter.

<Left Blank>

Chapter 4. SCHPLOT/SCHPRINT

4.1 SCHPLOT

4.1.1 Introduction

SCHPLOT performs foreground plotting of SCHDRAW-created drawings. SCHPLOT is the only SCHEMA III program used to produce plots. Plots cannot be created directly from SCHDRAW.

SCHPLOT operates in the foreground mode and thus is the only program that can be running while plots are being produced.

SCHPLOT normally outputs plot data directly to a communications port (specified in the configuration file). However, plot data can be redirected to a file or parallel port.

4.1.2 Plotter Communications

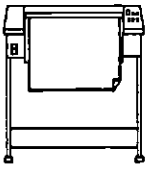
The **MODE** command *MUST* be invoked in order for SCHEMA to plot. Make sure there is not more than one **MODE** command in your **AUTOEXEC.BAT** that could conflict with each other.

Example: **MODE COM2:9600,N,8,1,P <Enter>**

There are two methods for which to plot: hardware-handshaking and software-handshaking. SCHEMA uses HARDWARE-HANDSHAKING. Just because your plotter works with a program like

AutoCAD, does not mean it works with SCHEMA. AutoCAD uses software-handshaking, or XON/XOFF. SCHEMA requires an extra hardware line called DTR(Data Terminal Ready) on the plotter to direct the computer when to start and stop transmission of data. This HARDWARE-HANDSHAKING cable must be configured as specified below in order for SCHEMA to plot. AutoCAD will work with this cable as well.

PLOTTER CABLE



9 PIN COM PORT

TRANSMIT DATA - PIN 3	RECEIVE DATA - PIN 3
RECEIVE DATA - PIN 2	TRANSMIT DATA - PIN 2
DSR - PIN 6	DATA TERMINAL READY - PIN 20
CTS - PIN 8	
GROUND - PIN 7	GROUND - PIN 7

25 PIN COM PORT

TRANSMIT DATA - PIN 2	RECEIVE DATA - PIN 3
RECEIVE DATA - PIN 3	TRANSMIT DATA - PIN 2
DSR - PIN 5	DATA TERMINAL READY - PIN 20
CTS - PIN 6	
GROUND - PIN 7	GROUND - PIN 7

SCHPLOT can scale a drawing to the size specified in the configuration file. For example, a virtual D-size drawing can be plotted on B-size paper, a virtual E-size drawing can be plotted on C-size paper, and so forth. SCHPLOT also allows you to position the drawing anywhere on the paper.

4.1.3 Invoking SCHPLOT

Like most SCHEMA III programs, SCHPLOT can be invoked from the Manager or directly from the DOS prompt. Chapter 2, *SCHEMA Program Manager*, contains detailed information on the Manager and the procedure to invoke SCHPLOT. Whether you use the Manager or DOS, the SCHPLOT command line is:

SCHPLOT DRAWING OPTION

where **DRAWING** is the name of the drawing to plot and **OPTION** specifies plot options.

DRAWING must specify the complete path and file name of the desired drawing. Do not include the drawing's file extension if you want all pages of the drawing to be plotted. If a single page is to be plotted, include the page's file extension. If more than one, but not all **DRAWING** pages are to be plotted, the desired pages can be specified. The three **DRAWING** formats are:

DRAWING - Plots all pages of drawing **DRAWING**, beginning with the first page, **DRAWING.P01**, followed by the second page, **DRAWING.P02**, etc. until all pages of **DRAWING** are plotted.

DRAWING.P01 - Plots only the single page **DRAWING.P01.**

DRAWING.P01,P07,P13 - Plots the page **DRAWING.P01**, followed by **DRAWING.P07** and **DRAWING.P13.**

In all cases, **DRAWING** includes the drawing's directory name/disk before the drawing name if the drawing does not reside in the current directory.

The size of the plot is dependent on the following parameters specified

in the appropriate configuration file.

NOTE

SCHPLOT first looks for a DRAWING.CFG file in the same directory as the DRAWING.Pxx file. If DRAWING.CFG is found, then all plot parameters are taken from it. If not found, SCHEMA III looks for SCHEMA.CFG in the current directory (not the drawing directory). If not there, then SCHEMA III looks for SCHEMA.CFG in the directory specified by the OMATION environment variable. If no configuration file is found, the program is aborted.

PLOTTER - Specifies the plotter driver to use. See later in this section for a list of supported plotters.

PLOTC - Specifies which serial communication port to use if output is routed directly to the plotter.

PXLEN - Specifies the plot area length available for plotting the x-direction of the drawing. This number is specified in mils (1000 mils = 1 inch). For example, 20400 specifies 20.4 inches. *Note that this number is the plot size, not the paper size.* Refer to your plotter manual to determine the maximum plot sizes for the paper you are using.

PYLEN - Specifies the plot area length available for plotting the y-direction of the drawing. This number is specified in mils (1000 mils = 1 inch). For example, 15400 specifies 15.4 inches. *Note that this number is the plot size, not the paper size.* Refer to your plotter manual to determine the maximum plot sizes for the paper you are using.

PXOFF - Specifies the offset from the plotter origin in the x-direction. This number is specified in mils (1000 mils = 1 inch). This allows the left/right edge of the drawing to be shifted on the physical page. Refer to your plotter manual to determine the location of the origin on the physical page. *Note that the origin for the larger HP plotters is the center of the physical page. In order to plot on the entire page, this parameter must be specified and must be negative* (e.g., -10200 moves the left edge of the drawing 10.2 inches to the left of the plotter origin).

PYOFF - Specifies the offset from the plotter origin in the y-direction. This number is specified in mils (1000 mils = 1 inch). This allows the top/bottom edge of the drawing to be shifted on the physical page. Refer to your plotter manual to determine the location of the origin on the physical page. *Note that the origin for the larger HP plotters is the center of the physical page. In order to plot on the entire page, this parameter must be specified and must be negative* (e.g., -7700 moves the bottom edge of the drawing 7.7 inches down from the plotter origin).

PENW - Specifies the width of the pen in mils (1000 mils = 1 inch). Only pen plotters require this parameter. If the width of your pen is given in millimeters, then divide the value by .0254 and truncate the result to an integer. For example, if the pen width is .35 mm, then the size in mils is .35/.0254 = 13.779. Finally, when truncated, the result is 13.

To show how the page length and offset parameters can be used, assume you have a C-size piece of paper (22x17) and that the plotter origin is the center of the page. If you want to use the lower left quarter of the physical page and the plotter manual states that the total plot area is 20.4 by 15.4 inches, then the following parameters are required:

PXLEN = 10200	(one half x-direction)
PYLEN = 7700	(one half y-direction)
PXOFF = -10200	(move to left hand edge)
PYOFF = -7700	(move to bottom edge)

The plot can be placed anywhere on the page as long as **PXLEN** + **PXOFF** does not exceed the right-hand edge of the page and **PYLEN** + **PYOFF** does not exceed the top edge of the page. If the page parameters specified are invalid for the physical page loaded, the plot output is clipped (i.e., not drawn).

Be aware that **PXLEN** and **PYLEN** always specify the plot area available for the x-direction and y-direction of the *drawing*, respectively. Likewise, **PXOFF** and **PYOFF** affect the location of the left edge and bottom edge of the drawing, respectively. This becomes important if the plotter driver supports the rotate option. Using the above example, but assuming rotated output is desired, then the following parameters are required:

PXLEN = 7700	(total length available for x-direction is 15.4 inches)
PYLEN = 10200	(total length available for y-direction is 20.4 inches)
PXOFF = -7700	(move left edge of drawing to bottom of paper)
PYOFF = 0	(bottom edge is at origin)

It is important to note that SCHPLOT always places the lower left corner of a drawing at the plotter origin (refer to figure 4-1). Values of the x-coordinate increase as you move left to right on the drawing; values of the y-coordinate increase as you move bottom to top on the drawing. Using this information along with your plotter manual, you can determine the effects of changing **PXOFF** and **PYOFF**.

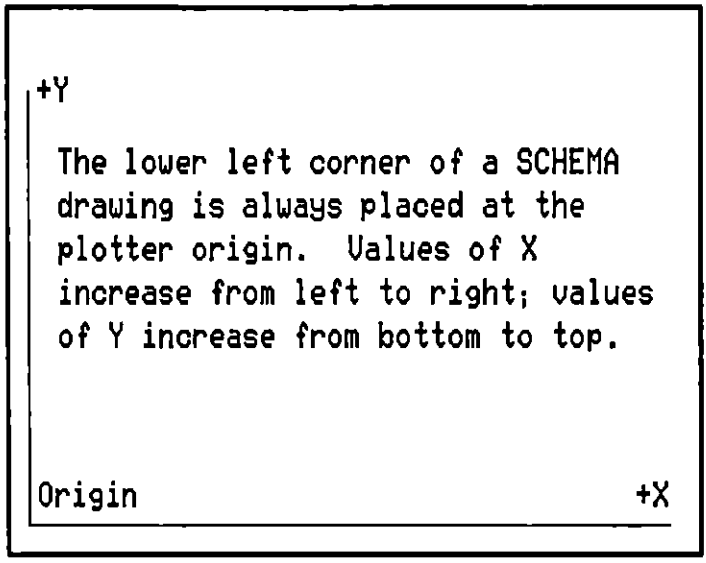

+Y
The lower left corner of a SCHEMA drawing is always placed at the plotter origin. Values of X increase from left to right; values of Y increase from bottom to top.
Origin
+X

Figure 4-1. Plotter Coordinates

Plot **OPTIONs** must be preceeded by a "/" and separated by one or more spaces. The following **OPTIONs** are supported:

/C - This option allows the user to change paper in the plotter. The following message is displayed before plotting each page:

Press <Enter> when plotter is ready

The **<Enter>** key must be pressed before any data is output to the

plotter. This option is ignored if the O option is specified. All plotter drivers support this option.

/E - This option causes the plotter to issue an "eject" command to the plotter when it finishes a plot. This is useful on plotters which support continuous feed mechanisms.

/P - This option causes all parts in a drawing to be defined to the plot driver prior to drawing the parts. Refer to Appendix E, *Hardware Requirements*, to determine how this option is used for the particular plot driver you are using. If this option is specified and the selected driver does not support the parts feature, the message appears:

Selected plotter doesn't support parts output - option ignored
This option is only for the DXF driver to output parts as DXF blocks.

/O - This option causes the plot output to be written to a file instead of directly to the plotter, or to a different COM or LPT port. One file is created for each drawing page plotted. The name of the file created is the same as the drawing name except the P in the extension is replaced with a S. For example, if pages, SAMPLE.P01 and SAMPLE.P03, are plotted using this option, then the files, SAMPLE.S01 and SAMPLE.S03, are generated. All plotter drivers support this option.

By default, the output files are placed in the same directory as the drawings. If you want the output to be placed on a different directory or drive, then use the /O option as follows: **/O=D:PATH**

where D: is optional and specifies the drive to place the files (required if different from the drive the file drawings are stored) and PATH specifies the full path of the directory where the files are to be placed. For example, if all plot output is to be placed on the same drive as the

drawings but in the PLOT directory, then the following would be used:

/O=\PLOT

A trailing backslash is not required, but does not cause an error. For example, /O=\PLOT\ and /O=\PLOT are equivalent. Any valid DOS drive and pathname can be used.

To output to a different COM or LPT port then enter: **/O=LPT2** or **/O=COM1.**

If a file or alternate port cannot be opened, the program is aborted and the following message is generated:

Error opening output file filename - Abort

If SCHPLOT is invoked without the O option, then the PLOTC parameter in the configuration file must specify the communications port to use. Furthermore, a DOS MODE command must be entered before plotting can begin. For example, if your plotter is connected to communications port 2 (PLOTC=2) and the plotter is configured for 9600 baud, no parity, 8 data bits and 1 stop bit, the following MODE command must be used:

MODE COM2:9600,N,8,1,P

These commands are typically placed in your AUTOEXEC.BAT file. For more information, consult your DOS USER'S MANUAL.

/R - This option causes the plot to be rotated 90-degrees (i.e., the x-direction of the drawing is plotted on the y-direction of the paper). If this option is specified and the selected plot driver does not support the rotate option, the following message is displayed:

Selected plotter doesn't support rotate - option ignored

Examples of SCHPLOT command lines are as follows:

SCHPLOT MEMORY /C <Enter>
(Plots all pages of drawing **MEMORY** and displays the **"Press <Enter> when plotter is ready"** message before outputting each page to the plotter.)

SCHPLOT \DRAWING\MEMORY.P03 <Enter>
(Plots only page 3 of drawing **MEMORY** from directory **\DRAWING**.)

SCHPLOT MEMORY.P01,P03 /O <Enter>
(Creates plot files **MEMORY.S01** and **MEMORY.S03** in the same directory.)

SCHPLOT MEMORY /R <Enter>
(Plots all pages rotated 90-degrees.)

4.1.4 Running SCHPLOT

After SCHPLOT loads, the following message is displayed:

SCHEMA Plotter Driver Version X.XX
Copyright Omation, Inc. 1989

(where: **X.XX** is the current version number.).

SCHPLOT finishes loading and, if the C option is specified, displays the following message when it is ready to begin plotting:

Press <Enter> when plotter is ready

When the necessary paper and pen are in the plotter, press **<Enter>**. SCHPLOT will plot the first page. When multiple pages are plotted, the above message is displayed between each page to allow time to change plotter paper.

The following message is displayed as each page is plotted:

Plotting DRAWING.PXX

where **DRAWING.PXX** is the current page being plotted.

To abort SCHPLOT press **<Esc>**. The plot will terminate immediately. Control is returned to the Manager or DOS when plotting is finished.

4.1.5 PLOTTER PROBLEM CHECKLIST

1. The **PLOTTER** variable in the configuration file must contain the entry for the plotter to be used.
2. Verify the plotter cable is configured as the pin outs specify.
3. Make sure the plotter is attached to the same com port as the **PLOTC** variable in the configuration file.
4. Make sure the plotter communication parameters are the same as those specified via the **MODE** command (baud rate, data bits, parity and stop bits).

The following discusses each plotter type supported by SCHEMA III. Names in parentheses are used in the configuration file to specify plotter type. Any newly-supported plotter types are listed in the READ.ME file on the SCHEMA III Program disk.

4.1.6 HEWLETT-PACKARD (HP)

This plotter type supports most plotters that use the Hewlett-Packard HP-GL plotter language. Note that the origin for the larger HP plotters (e.g., 758x) is located at the center of the page. Thus, if you wish to force the plot to the left hand corner, negative offsets for the configuration variables **PXOFF** and **PYOFF** probably need to be specified.

For example, let's say that we are plotting to a B size sheet and the maximum plotting area for this paper is 16000 mils by 10000 mils. The **PXLEN** is 16000 and the **PYLEN** is 1000. Therefore, to force the plot to the lower left hand corner the offsets would have to be:

PXOFF= -8000 and **PYOFF** = -5000.

This driver does not support the /P (parts) option. The /R (rotate) option causes a R090 command to be sent to the plotter during initialization. Be sure to refer to your plotter manual to determine the effects of specifying the **PXOFF** and **PYOFF** parameters for both standard and rotated plots.

4.1.7 HOUSTON INSTRUMENT (HI)

This plotter type supports most plotters that use the Houston Instrument DM/PL plotter language. This driver does not support the /P (parts) option. The /R (rotate) option is used to specify the chart size. If /R is not specified then Large Chart is assumed. If /R is specified, Small Chart is assumed. Note that you can effectively rotate the plot 90-degrees by specifying a chart size different than the one actually loaded. For example, the HI DMP-61 requires a "D" size page to be loaded in Large Chart format (chart axis is larger than pen axis). In this mode, the plot origin is on the right side of the page with the Y-axis being along the pen axis. If the /R option is specified in this case, the plot is drawn with the origin on the left side of the page and the x-axis being along the pen

axis. Also, note that the effects of **PXOFF** and **PYOFF** change depending on the /R option. Refer to your plotter manual to determine the x- and y-axis orientation for each chart type.

4.1.8 CALCOMP (CP)

This plotter type supports the CALCOMP series of plotters. The driver name is **CP**. This is to be placed in the **PLOTTER=** field in your DRAWING.CFG. If you do not have a .CFG for your drawings, put this in the SCHEMA.CFG file.

The hardware should be set up in the following manner:

- **Language - PCI**
- **Set the baud rate the same as in the MODE command (typically 9600)**
- **Byte length should be 8**
- **Stop bits should be 1**
- **Parity None**
- **Handshake should be Direct Control (XON/XOFF set to OFF)**
- **Sync Character is 2**
- **The number of Sync Characters is 1**
- **End of Message is 3**
- **Checksum should be disabled**

IMPORTANT!! Check the Plotter Interface for your plotter:

If the plotter interface is configured as Data Communications Equipment (DCE), then you must use a straight through cable. DO NOT USE THE CABLE DESCRIBED IN THE SCHEMA III MANUAL. The critical lines that must be straight through are:

2 - Transmit Data
3 - Receive Data
5 - CTS Clear to Send
7 - Ground

If the plotter is configured for Data Terminal Equipment (DTE), use the cable described previously.

4.1.9 POSTSCRIPT (PS)

This plotter type actually generates output in the PostScript interpretive graphics description language for printing on raster devices, such as the Apple LaserWriter, which have a PostScript language interpreter resident in the printer. PostScript output may also be printed on conventional matrix or laser printers with the use of software based interpreters such as GoScript from LaserGo, Inc. This driver does not support the /P (parts) option. The /R (rotate) option causes the plot to be rotated 90-degrees. When /R is not specified, the plotter origin is the upper left corner of the paper. The x-coordinates increase down the page and the y-coordinates increase across (left-to-right) the page. When /R is specified the plotter origin is the lower left corner of the paper. The x-coordinates increase across (left-to-right) the page and the y-coordinate increase up (bottom-to-top) the page. Use the /O option to redirect to a file if necessary.

Example:

Let's say that we have a schematic called, PROJECTX. We want to take the first page to PostScript. Use SCHCFG to load the config file, **PROJECTX.CFG** and make sure the **PLOTTER=** is PS. Check the X and Y lengths for scaling.

Use SCHPLOT to execute the following statement:

SCHPLOT PROJECTX.P01 /O /R <Enter>

The /O switch is optional and redirects the output to the file **PROJECTX.S01.**

The /R is optional and will rotate the schematic 90 degrees.

NOTE: *If you are sending this file through a COM port then you must use the cable specified.*

4.1.10 DRAWING EXCHANGE FORMAT (DXF)

This plotter driver actually generates output in the DXF format which can then be read by **AutoCAD** (using **DXFIN**) and other programs. The /R (rotate) option is not supported. The /P (parts) option is supported and, if specified, causes all of the parts used in a drawing to be defined in the DXF block format. This allows SCHEMA parts to be moved and placed as a single entity even though they may be made up of several lines, arcs, text, etc. These blocks also carry the SCHEMA description and parameter fields which allow them to be extracted via the **AutoCAD** (ATTEXT) command using the SDF option. If the /P option is not specified, then parts are drawn but not defined as blocks. Use the /O option to redirect to a file for transferal to **AutoCAD.**

Example:

Let's say we have a schematic called, PROJECTX. We want to take the first page to AutoCAD. Use SCHCFG to load the config file, **PROJECTX.CFG** and make sure the plotter is setup for **DXF.** Check the X and Y lengths for scaling.

Use SCHPLOT to execute the following statement:

SCHPLOT PROJECTX.P01 /O /P

The /O switch is **required** and redirects the DXF output to the file **PROJECTX.S01.**

The /P switch is optional and will group the schematic parts into BLOCKS in DXF.

Next, you must rename the file **PROJECTX.S01** to **PROJECTX.DXF** for **AutoCAD** input.

Invoke **AutoCAD** and select the option, **DXFIN.** Read the **PROJECTX.DXF** file into **AutoCAD.**

4.1.11 Customizing Pages For A Particular Plotter

Since plot drivers scale the drawing to fit on a particular size of paper, there is very little reason to create a custom sized page for plotting. However, you may notice the plot output may completely fill one direction of the paper while leaving an unused area in the other direction. This occurs because the plotter always outputs a drawing using an aspect ratio of 1.00 (i.e., the drawing is scaled the same amount in both the x- and y-directions.

For example, suppose you are plotting on a piece of paper where **PXLEN** = 32800 and **PYLEN** = 20800. If we use a standard SCHEMA D-size page (1600 x 1216), the scale factor is calculated as follows:

32800/1600 = 20.5
20800/1216 = 17.1

Since the scale factors for the x- and y-direction are different, SCHPLOT chooses the smaller of the two to insure the plot will fit on the paper. Thus, in this example, the x-direction of the page will have an unused area. To make the plot use more of the paper, we can adjust either the x or y dot count. Let's assume we want to decrease the page size instead of increase it. This means we need to change the number of y dots in the page such that $20800/y = 20.5$.

After solving, $y = 1014$. Remember, however, that the number of y dots must be a multiple of 16. The closest multiple of 16 to 1014 is 1008. Therefore, we use SCHPAGE to create a 1600 x 1008 page which, when plotted on the paper size discussed above, utilizes the paper very well.

SCHPAGE is discussed in *Chapter 8 - Utilities*.

4.2 SCHPRINT

4.2.1 Introduction

SCHPRINT performs foreground printing of SCHDRAW-created drawings. Foreground operation occurs when the PC is being used 100-percent for one operation, i.e., no other programs are running. This is in contrast to SCHDRAW's **FILE-PRINT** command which is normally used to do background printing. Background printing allows SCHDRAW (or any other DOS program) to run at the same time printing is done. However, in order to do background printing, a temporary print file is created on disk. This file can be up to 600K bytes long. Thus, if there is insufficient disk space for a file (or files) of this size, it is impossible to use SCHDRAW in the background print mode. SCHPRINT prints multiple pages of a drawing without user intervention.

Pages can be printed with various printer resolutions and at any zoom level. Refer to Appendix E, *Hardware Requirements*, to determine what print modes (resolutions) are available for the printer you are using. Pages can also be printed in one of four zoom levels: 1:1, 2:1, 4:1 and 8:1. Printing at the 2:1 zoom level results in a printout half the size in both the horizontal and vertical directions. Printing at a zoom level other than 1:1 can be useful to simply check a printout. The various printer resolutions are provided for selection of the most appealing printout.

As an example, the EPSON Wide-Carriage printer has three modes. Each mode offers a different resolution as follows:

A Mode: 120 x 108 dots per inch
C Mode: 60 x 72 dots per inch
D Mode: 60 x 54 dots per inch

The mode which generates the "best" output is often a matter of taste. However, the amount of paper required to print a drawing increases as the print density (dots per inch) decreases.

To determine how many physical pages are required to print a drawing page, use the SCHPAGE utility to get the number of dots in both the x- and y-directions of the page. For example, the **BPAGE2.P99** page is 1600 x 1216. If this page is printed with the Epson driver (mode A) in landscape mode, then 1216 dots need to be printed in the x-direction. At 120 dots per inch, this requires 1216/120 = 10.133 inches. Thus, on a wide carriage printer, only one width of paper is required. On a narrow carriage printer, the drawing is automatically split into multiple sheets; these sheets can then be "cut and pasted" to get the full page. Note when multiple sheets are required, dots near the borders are duplicated on the next page to make it easier to align the paper.

SCHPRINT does not require disk space to print a file. Also, SCHPRINT can print multiple files unattended making it ideal when leaving your PC for long periods of time (such as overnight).

NOTE

Appendix E, Hardware Requirements, contains information regarding printer types supported by SCHEMA III. Before proceeding, read the information in Appendix E that applies to your printer.

4.2.2 Invoking SCHPRINT

Like most SCHEMA III programs, SCHPRINT can be invoked from the Manager or directly from the DOS prompt. Chapter 2, *SCHEMA Program Manager*, contains detailed information on the Manager and the procedure to invoke SCHPRINT. Whether you use the Manager or DOS, the SCHPRINT command line is:

SCHPRINT DRAWING OPTION

where **DRAWING** is the name of the drawing to print and **OPTION** specifies print options.

DRAWING must specify the complete path and file name of the desired drawing. Do not include the drawing's file extension if you want all pages of the drawing to be printed. If a single page is to be printed, include the page's file extension. If more than one, but not all **DRAWING** pages are to be printed, the desired pages can be specified. The three **DRAWING** formats are:

DRAWING - Prints all pages of drawing **DRAWING**, beginning with the first page, **DRAWING.P01**, followed by the second page, **DRAWING.P02**, until all pages of **DRAWING** are printed.

DRAWING.P01 - Prints only the single page **DRAWING.P01**.

DRAWING.P01,P07,P13 - Prints the page **DRAWING.P01**, followed by **DRAWING.P07** and **DRAWING.P13**.

In all cases, **DRAWING** includes the drawing's directory name/disk before the drawing name if the drawing does not reside in the current directory.

Print **OPTIONs** must be preceeded by a "/" and separated by one or more spaces. The following **OPTIONs** are supported:

/m - This option specifies the mode to use. Replace **m** with the mode character you desire (e.g., Epson supports modes A, C and D.). Default is mode A for all drivers.

/n - This option specifies the zoom level to print. Replace **n** with 2, 4 or 8 (2:1, 4:1 or 8:1). Default is **1:1.**

/O - This option causes the print output to be written to a file instead of directly to the printer, or to a different COM or LPT port. One file is created for each drawing page plotted. The name of the file created is the same as the drawing name except the **P** in the extension is replaced with a **T**. For example, if pages, SAMPLE.P01 and SAMPLE.P03, are printed using this option, then the files, SAMPLE.T01 and SAMPLE.T03, are generated.

By default, the output files are placed in the same directory as the drawings. If you want the output to be placed on a different directory or drive, then use the /O option as follows:

/O=D:\PATH

where **D:** is optional and specifies the drive to place the files (required if different from the drive the file drawings are stored) and **PATH** specifies the full path of the directory where the files are to be placed. For example, if all print output is to be placed on the same drive as the drawings but in the **PRINT** directory, then the following would be used:

/O=\PRINT

A trailing backslash is not required, but does not cause an error. For example, /O=\PRINT\ and /O=\PRINT are equivalent. Any valid DOS drive and pathname can be used.

If a file cannot be opened, the program is aborted and the following message is generated:

Error opening output file filename - Abort

To output to a different COM or LPT port then enter: **/O=LPT2** or **/O=COM1.**

/R - This option generates a portrait printout, where the x-direction of the drawing is printed on the x-direction of the paper. All printer drivers are capable of printing a drawing in one of two orientations. The default orientation generates a "sideways" printout, where the x-direction of the drawing is printed on the y-direction of the paper.

Examples of SCHPRINT command lines are as follows:

SCHPRINT MEMORY /C /2 <Enter>
(Prints all pages of drawing **MEMORY** in the C mode
and 2:1 zoom level)

SCHPRINT \DRAWING\MEMORY /D <Enter>
(Prints all pages of drawing **MEMORY** from directory **DRAWING** in
the full-size mode)

SCHPRINT MEMORY.P03 <Enter>
(Prints page 3 of drawing **MEMORY** in the normal mode)

SCHPRINT MEMORY.P01,P03,P05 <Enter>
(Prints pages 1, 3, and 5 of drawing **MEMORY** in the normal mode).

SCHPRINT MEMORY.P01 /R /2 <Enter>
(Prints page 1 of drawing **MEMORY**
at the 2:1 zoom level in portrait format)

SCHPRINT MEMORY /R /C /O
Prints all pages of drawing **MEMORY** at the 1:1 zoom level using Mode C in portrait format to files with a T in the extension.

4.2.3 Running SCHPRINT

After SCHPRINT is loaded, the following message is displayed:

SCHEMA PRINTER DRIVER Version X.XX
Copyright Omation, Inc 1989

(where: **X.XX** is the current version number.).

SCHPRINT finishes loading and prints the first drawing. The following message is displayed as each page is printed:

PRINTING DRAWING.PXX

where **DRAWING.PXX** is the current page being printed.

No further user intervention is required. If multiple drawings are printed, an automatic form feed is sent to the printer between each page.

To abort SCHPRINT, press **<Esc>**. SCHPRINT aborts after the current print line finishes.

Control is returned to the Manager or DOS after printing is finished.

4.2.4 Customizing Pages For A Particular Printer

When pages are printed, no scaling takes place except for the 2:1, 4:1 and 8:1 zoom levels allowed. In other words, the printer drivers do not scale a drawing to make it fit on a specific piece of paper. If you always use a particular printer, or, have a drawing which must fit on a single physical sheet of paper, then you may want to use SCHPAGE to create a custom page size.

To determine the page size, simply multiply the width of the paper by the number of dots per inch the printer prints in the x-direction. This gives the number of dots that can be printed in the x-direction. Do the same arithmetic for the y-direction.

For example, assume you have a narrow carriage EPSON printer and want to create a page that will fit on a single sheet of letter-size paper using mode A (120 x 108 dpi). The number of dots that can be printed are:

Width = (8")(120 dpi) = 960 dots
Length = (10")(108 dpi) = 1080 dots

If you print using the portrait (/P) option, then the page size you need to create is 960 x 1080. If you do not print with the portrait option, then the page size you need is 1080 x 960. Note that the numbers you calculate may have to be modified to meet the requirements of SCHPAGE (x-dots must be a multiple of 32, y-dots must be a multiple of 16). For example, the 1080 x 960 calculated above is not a valid page size (1080 is not a multiple of 32); thus, the page you must actually create is 1056 x 960.

4.2.5 Interfacing to Word Processors using TIFF

The output for TIFF is supplied from the printer driver, TIFF.DRV. This format can be read by **PageMaker, Word, WordPerfect, Ventura,** and others.

Example:

Let's say we have a schematic called, PROJECTX. We want to take the first page to the TIFF format. Use SCHCFG to load the config file

SCHEMA.CFG and make sure the printer is setup for TIFF.DRV. Remember, printer drivers are *ONLY* taken from the SCHEMA.CFG.

Use SCHPRINT to execute the following statement:

SCHPRINT PROJECTX.P01 /O /R <Enter>

The /O switch is required and redirects the TIFF output to the file PROJECTX.T01.

The /R switch rotates the output 90 degrees. For most word processors this switch is needed to have the image read in a portrait mode.

Now you can load the TIFF file into your word processor.

Note: If you wish to cut a region from a schematic to output to a file, Use **EDIT-REGION-SAVE** and give the region an extension of (.P01).

Chapter 5. SCHCFG

5.1 INTRODUCTION

SCHCFG lets you create/edit SCHEMA III configuration files. There are two types of SCHEMA III configuration files: DRAWING.CFG and SCHEMA.CFG.

One DRAWING.CFG file (where DRAWING is the name of the associated drawing) can be created for each schematic drawing set you draw. The DRAWING.CFG file contains the associated drawing's custom hardware configuration, title block information, object library assignments, and post-processor formatting information. If a DRAWING.CFG file is not created for a given drawing set, the default information contained in the global SCHEMA.CFG file is used by the drawing set.

SCHEMA.CFG serves as the default configuration file for schematic drawing sets which do not have a DRAWING.CFG file and as a template for creating new DRAWING.CFG files.

5.2 CONFIGURATION FILE FORMAT

The SCHEMA III configuration files are standard ASCII files. They can be read or printed using standard DOS routines (TYPE, PRINT, COPY, etc.). Also, the configuration files can be modified via any text editor instead of using SCHCFG, if desired.

The configuration files consist of a series of text lines. Each line

contains a variable, followed by an equal sign (=) and parameter, such as:

GRAPH=EG (EG = EGA)

This example sets the graphics adapter type (**GRAPH**) variable in the SCHEMA.CFG file to work with the IBM Enhanced Graphics Adapter (**EGA**).

5.2.1 Variable Description

Configuration file variables are divided into the following five categories:

a. Equipment or hardware configuration
b. SCHDRAW configuration
c. Schematic title block
d. Object library
e. SCHPOST bill of materials

When a SCHEMA III program loads a drawing, the corresponding drawing configuration file is also loaded. SCHEMA III first looks for **DRAWING.CFG** in the directory that **DRAWING** (drawing file name) is located. If **DRAWING.CFG** is not found, SCHEMA III tries to find **SCHEMA.CFG** in the **DRAWING** directory. If **SCHEMA.CFG** is not found in the **DRAWING** directory, **SCHEMA.CFG** file in the Environment Variable directory (typically the \SCHEMA directory) is loaded. Refer to Appendix A, *INSTALLING SCHEMA III* for details on setting the environment variable.

The equipment variables configure SCHEMA III to work with the hardware your PC uses and the output requirements of the particular DRAWING or project. **GRAPH** assigns the graphics adapter type and display colors. **PRINT** assigns printer or printer output type. **PLOTTER**

defines the plotter or plot output configuration along with the additional equipment variables (**PLOTC, PXLEN, PYLEN, PXOFF, PYOFF** and **PENW**).

The SCHDRAW configuration variables (**PAN, RPTX, RPTY, XY, BACKUP, ZOOM, CGRID, VGRID, WIHDR, DIAG,** and **MOUSE**) are used to set the load-time values of these parameters in the Schematic Editor when SCHDRAW is invoked. **PAN** set the Drawing Window panning speed. **RPTX** and **RPTY** set the repeat X- and Y-offsets for the SCHDRAW **DRAW-REPEAT** command. **XY** turns the Cursor Coordinate Window on and off. **BACKUP** determines number of minutes between auto-backups in SCHDRAW. **ZOOM** determines what ZOOM level is preferred as the default. **CGRID** turns the Cursor Grid on or off. **VGRID** turns the Visual Grid on or off. **DIAG** allows diagonal lines to be drawn. **WIHDR** turns the Wire Header feature on or off. **MOUSE** sets the mouse horizontal and vertical sensitivity.

Schematic title block variables **F1** through **F8** are used to preassign schematic title block information and the titles printed across the top of most of the SCHPOST outputs. See paragraph 3.3.2 and the SCHDRAW **NOTES-TEXT** command description in Chapter 3 for additional information on these variables. Typically, variable **F1** is the schematic name, **F2** the schematic drawing number, **F3** the revision number (or name), and **F4** the revision date. **F5** is the last page number of the drawing, and **F6, F7,** and **F8** are undefined. These variables provide a means of assigning title block information once, with all schematic page title blocks and associated SCHPOST printouts containing the same information.

Object library assignment variables **LIBR1** through **LIBR9** assign object libraries for use by the associated schematic. The field of each object library assignment line contains the DOS directory (if the library is not in the current directory) and file name of the SCHEMA III library file.

These nine object libraries are the only libraries the applicable schematic can access. The **INSERT** and **DELETE** commands are used to insert and delete object library entries. **PROMPT** defines a load-time prompt for floppy disk operation. If a **PROMPT** line is included in the SCHEMA.CFG file, a prompt is displayed in floppy disk systems to inform users when a disk swap is necessary. Hard disk systems do not use the **PROMPT** variable.

SCHPOST bill of materials assignment variables **BOM1** through **BOM3** assign header (BOM title), column identifiers (column titles), and field makeup information for use by the SCHPOST Bill of Materials post-processor. Text assigned to the **BOM1** variable is used as the header (title) printed along the top of each BOM page. Text assigned to the **BOM2** variable is used as the column identifier line, also printed along the top of each BOM page. Values assigned to the **BOM3** variable determine BOM field types, lengths, and justification information. These fields make up each BOM information line. Refer to Chapter 7, *SCHPOST*, for additional information on these lines.

NOTE

Due to the permissible length of each BOM variable (140 characters to accomodate wide carriage printer output), SCHCFG presents the three BOM variables as 6 70-character variables representing the halves of the BOM variables.

5.3 INVOKING SCHCFG

Like most SCHEMA III programs, SCHCFG can be invoked from the Manager or directly from the DOS prompt. Chapter 2, *SCHEMA Program Manager*, contains detailed information on the Manager and the procedure to invoke SCHCFG. Whether you use the Manager or DOS, the SCHCFG command line is:

SCHCFG FILE

where FILE is the optional configuration file to load. The optional .CFG extension on FILE is assumed if an extension is not specified. If FILE is not specified, SCHCFG tries to load the SCHEMA.CFG file from the current directory. If SCHEMA.CFG is not found, it will be loaded from the "environment variable" directory. If FILE is specified, the complete path and name of the desired configuration file must be included unless it resides in the directory specified by FILE. After <Enter> is pressed, SCHCFG is invoked.

NOTE

SCHEMA III programs first look for the SCHEMA.CFG in the current directory. If not there, then SCHEMA III looks in the directory specified by the OMATION environment variable. The only parameters that it retrieves are the graphics and printer driver variables (GRAPH= and PRINT=). Then SCHEMA III searches for DRAWING.CFG in the same directory as the drawing to get all the remaining variables. If no DRAWING.CFG file is found then the rest of the variables will be taken from the SCHEMA.CFG file.

5.4 RUNNING SCHCFG

When SCHCFG is invoked, the screen will display the active file name and the Main Configuration Menu, shown in figure 5-1, which lists five other sub-menus. Each sub-menu lets you create/modify a different group of configuration file variables. Figure 5-2 through 5-6 show these menus. When viewing a subset of configuration variables, the actual variable names and their current settings are listed in the upper portion of the screen. This text can then be modified. Highlight the variable you wish to edit. When <Enter> is pressed, the text line appears. Edit the

contents of the variable. Press **<Enter>** again to finish editing. The updated text is placed in the upper portion of the screen next to the selected variable name. If the variable does not need to be edited, the text line can be exited by pressing **<Esc>** repeatedly. The **EXIT** command (on each menu) or **<Esc>** key allows quick menu exits.

NOTE

*Within the Hardware sub-menu when selecting **GRAPH, PRINT** or **PLOTTER**, a selection screen will appear with a list of graphics adapters, printer drivers or plotters supported in the current software release. In this case, a third menu level allows direct access to the hardware variables for editing when DISPLAY/EDIT CURRENT SETTINGS is selected.*

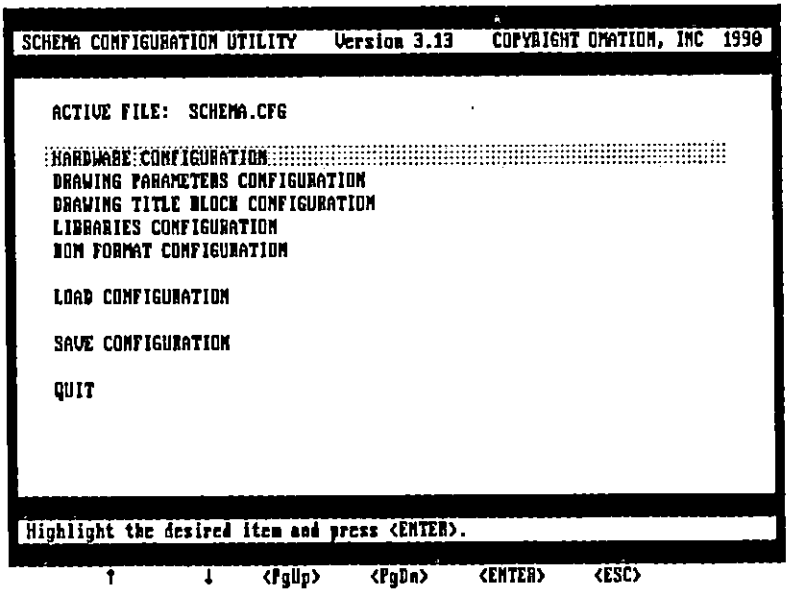


Figure 5-1. SCHCFG Main Configuration Menu

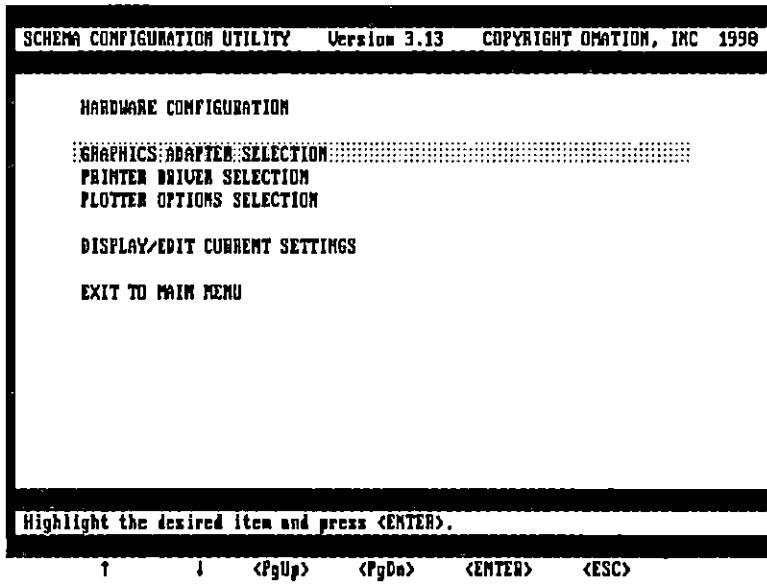

SCHEMA CONFIGURATION UTILITY Version 3.13 COPYRIGHT OMATION, INC 1990
HARDWARE CONFIGURATION
GRAPHICS ADAPTER SELECTION
PRINTER DRIVER SELECTION
PLOTTER OPTIONS SELECTION
DISPLAY/EDIT CURRENT SETTINGS
EXIT TO MAIN MENU
Highlight the desired item and press <ENTER>.
↑ ↓ <PgUp> <PgDn> <ENTER> <ESC>

Figure 5-2. SCHCFG Hardware Configuration Menu

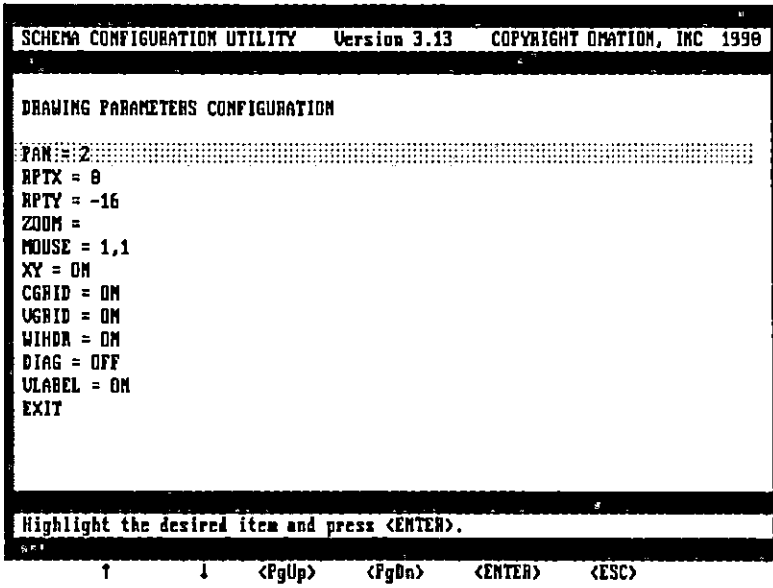


Figure 5-3. SCHCFG Drawing Parameters Configuration Menu

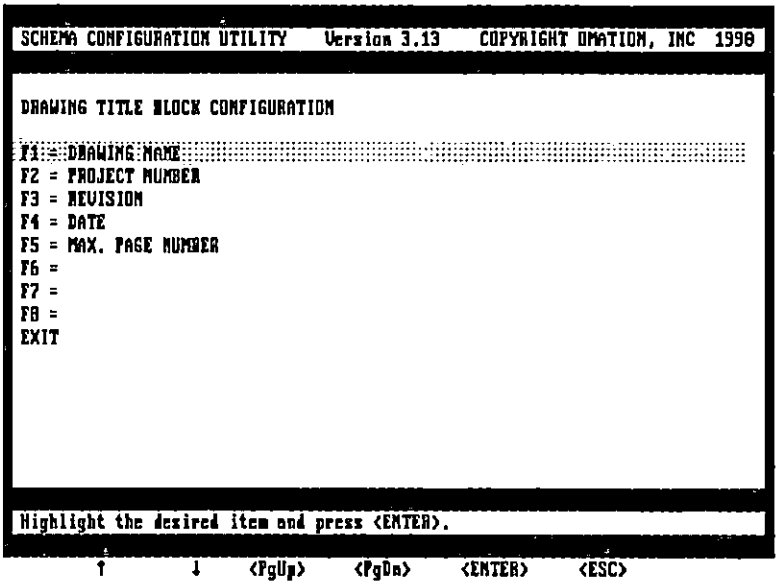


Figure 5-4. SCHCFG Drawing Title Block Configuration Menu

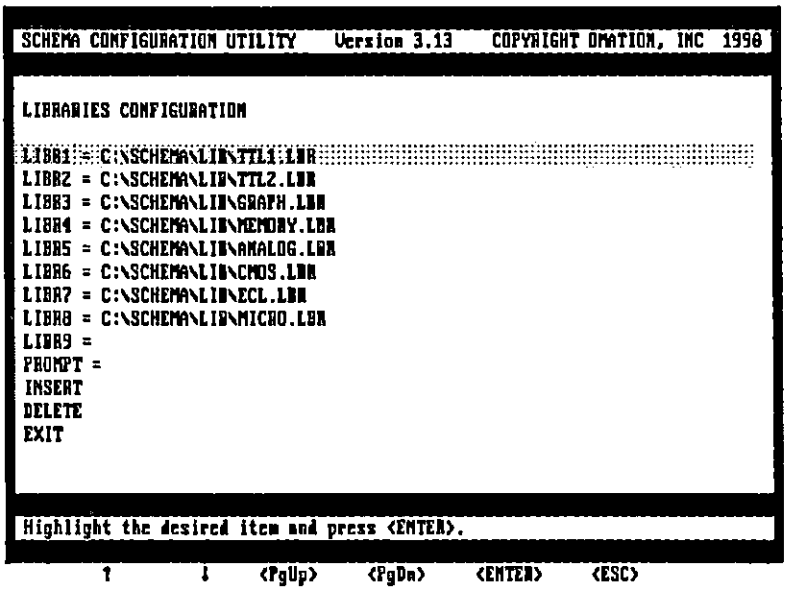


Figure 5-5. SCHCFG Libraries Configuration Menu

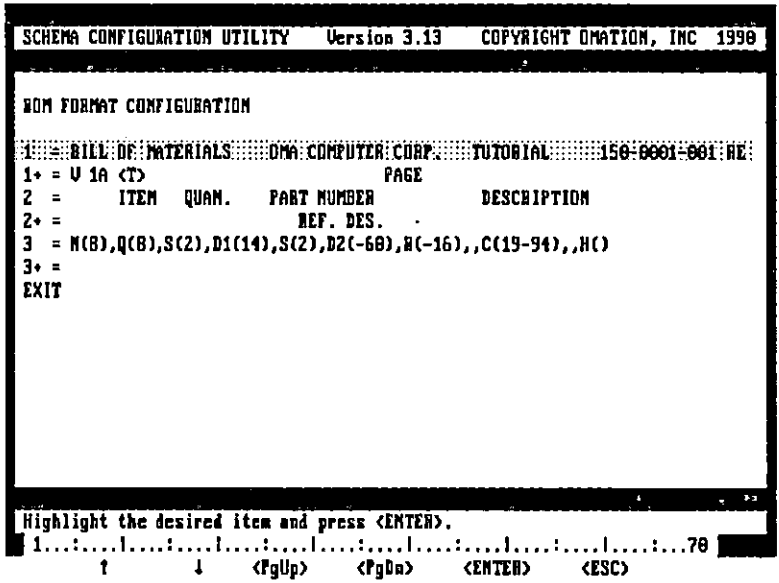

SCHEMA CONFIGURATION UTILITY Version 3.13 COPYRIGHT OMATION, INC 1990
BOM FORMAT CONFIGURATION
1 = BILL OF MATERIALS OMA COMPUTER CORP. TUTORIAL 150-0001-001 RE
1+ = V 1A <T> PAGE
2 = ITEM QUAN. PART NUMBER DESCRIPTION
2+ = REF. DES.
3 = N(8),Q(8),S(2),D1(14),S(2),D2(-68),N(-16),,C(19-94),,N()
3+ =
EXIT
Highlight the desired item and press <ENTER>.
1...:....|....:....|....:....|....:....|....:....|....:....|....:....|....:...70
↑ ↓ <PgUp> <PgDn> <ENTER> <ESC>

Figure 5-6. SCHCFG BOM Format Configuration Menu

A single line box is reserved below the menu commands for user prompts. When commands are invoked, user prompts are displayed requesting a specific entry or action.

As previously stated, the last line of each menu is the text line. All keyboard text is displayed in the text line. Appendix C covers text editing keyboard usage.

The Main Configuration Menu also provides access to the LOAD, SAVE and QUIT commands. The LOAD command loads a configuration file. The SAVE command saves the current configuration information to the specified file. Both LOAD and SAVE allow the user to edit the filename before being invoked. If a full path filename is not given, the current directory is assumed. If the LOAD filename contains DOS wildcard characters (* ?), the drawing utility is invoked to browse and select files matching the filename. The QUIT command is used to exit SCHCFG and return to the Manager or DOS without saving the configuration file. The Hardware Configuration Menu provides access to the PRINT, GRAPH and PLOTTER (PLOTC, PXLEN, PYLEN, PXOFF, PYOFF and PENW) variables.

The Drawing Parameters Configuration Menu allows you to set the SCHDRAW PAN, RPTX, RPTY, XY, ZOOM, CGRID, VGRID, DIAG, WIHDR and MOUSE load-time variables.

The Title Block Menu lets you modify/create the F1-F8 variables.

The Library Menus let you modify/create the LIBR1-LIBR9 and PROMPT variables.

The Bill of Materials Menu lets you modify/create the BOM1-BOM3 variables.

5.5 COMMAND DESCRIPTIONS

This paragraph contains detailed descriptions of SCHCFG commands. Each command is described as follows:

- **Use:** Defines reason for use

- **Operation:** Lists procedure to use command

- **See Also:** Identifies related commands/information, if applicable

- **Notes:** Identifies special conditions, limitations, etc. associated with command, if applicable.

SCHCFG Menu Command Structure

ACTIVE FILE: XXXXXXXX.CFG

Main Configuration Menu:

Hardware Configuration:

*Graphics Adapter Selection:

List of graphics adapters

*Printer Driver Selection:

List of printer drivers

Plotter Options Selection:

List of plotters

Display/Edit Current Settings:

***GRAPH**	**PYLEN**
***PRINT**	**PXOFF**
PLOTTER	**PYOFF**
PLOTC	**PENW**
PXLEN	**EXIT**

Exit To Main Menu

* Available only with SCHEMA.CFG

Drawing Parameters Configuration:

PAN	**CGRID**
	VGRID
RPTX	**WIHDR**
PRTY	**DIAG**
XY	**MOUSE**
ZOOM	**EXIT**

Drawing Title Block Configuration:

F1 - F8
EXIT

Libraries Configuration:

LIBR1 - LIBR9
PROMPT
INSERT
DELETE
EXIT

BOM Format Configuration:

BOM1 - BOM3+
EXIT

LOAD Configuration

SAVE Configuration

QUIT

BACKUP (DRAWING PARAMETERS CONFIGURATION MENU)

Use: To set the SCHDRAW auto-backup interval.

Operation: When selected the **BACKUP** variable lights. Press **<Enter>** and the following prompt is displayed:

Editing the configuration parameter BACKUP.

The default value is 5. Type in any integer value with the units being minutes. A value of 0 will disable the auto-backup. The file updated is called, **BACKUP.P01.** SCHEMA will use the **OMATION=** environment variable to find the path to where to send the file. If the environment variable is not found then backup will be saved to the current directory.

See Also: Chapter 3, *SCHDRAW*

Notes: This parameter is only read from the SCHEMA.CFG file.

BOMx(+) (BOM FORMAT CONFIGURATION MENU)

Use: To assign the first or last 70 characters to the BOM configuration variables. See paragraph 7.4.2 for a detailed explanation of BOMx use (where x is 1, 1+, 2, 2+, 3, or 3+).

Operation: When selected the **BOMx** variable lights. Press **<Enter>** and the following prompt is displayed:

Editing configuration parameter BOMx.

When x is 1, 2 or 3, enter or edit the first 70 characters of the BOM variable followed by **<Enter>**. When x is 1+, 2+ or 3+, enter or edit the last 70 characters of the BOM variable followed by **<Enter>**.

See Also: Chapter 7, *SCHPOST*

CGRID (DRAWING PARAMETERS CONFIGURATION MENU)

Use: To turn the SCHDRAW invisible snap Cursor Grid (**C GRID**) on or off at load-time.

Operation: When selected the **CGRID** variable toggles between **ON** and **OFF.**

See Also: Chapter 3, *SCHDRAW*

Notes: The value assigned to **CGRID** can be changed from SCHDRAW using **CONFIG-C GRID.**

DELETE (LIBRARIES CONFIGURATION MENU)

Use: To delete an object library entry from the Library Menu.

Operation: When selected the **DELETE** command lights. Press **<Enter>** and the **DELETE** command is activated. **DELETE** toggles between **ON** and **OFF**. When activated, **LIBR ON** appears next to **DELETE**.

Select the library you want to delete followed by **<Enter>**. The library is deleted and any libraries assigned to higher **LIBR** variables are moved up.

For example, if **LIBR4** is selected, press **<Enter>** and **DELETE** moves the library assigned to **LIBR5** to **LIBR4**, the library assigned to **LIBR6** is moved to **LIBR5**, and **LIBR6** is now blank.

See Also: **INSERT** command

Notes: **INSERT** and **DELETE** are mutually exclusive. Both commands cannot be active at once. Activating one de-activates the other.

DIAG (DRAWING PARAMETERS CONFIGURATION MENU)

Use: To enable or disable SCHDRAW diagonal lines at load-time.

Operation: When selected the DIAG variable toggles between ON and OFF.

See Also: Chapter 3, *SCHDRAW*

Notes: The value assigned to DIAG can be changed from SCHDRAW using CONFIG-DIAG.

EXIT

Use: To move from the current sub-menu to the Main Configuration Menu.

Operation: When selected the current menu is replaced with the Main Configuration Menu.

The Hardware Configuration sub-menus are nested two menus deep. To move up one menu level, select EXIT until the desired menu appears.

F1 (TITLE BLOCK CONFIG. MENU)

Use: To assign a **DRAWING NAME** to the drawing associated with this configuration file. The drawing name assigned to this variable is included in the SCHPOST header (title) and is also placed in the title block of the applicable drawing.

Operation: When selected the **F1** variable lights. Press **<Enter>** and the following prompt is displayed:

Editing configuration parameter F1.

Enter the desired drawing name (up to 54 alphanumeric characters) followed by **<Enter>** again.

F2 (TITLE BLOCK CONFIG. MENU)

Use: To assign a **DRAWING NUMBER** to the drawing associated with this configuration file. The drawing number assigned to this variable is included in the SCHPOST header (title) and is also placed in the title block of the applicable drawing.

Operation: When selected the **F2** variable lights. Press **<Enter>** and the following prompt is displayed:

Editing configuration parameter F2.

Enter the desired drawing number followed by **<Enter>** again.

F3 (TITLE BLOCK CONFIG. MENU)

Use: To assign a **DRAWING REVISION NAME and/or NUMBER** to the drawing associated with this configuration file. The revision name or number assigned to this variable is included in the SCHPOST header (title) and is also placed in the title block of the applicable drawing.

Operation: When selected the **F3** variable lights. Press **<Enter>** and the following prompt is displayed:

Editing configuration parameter F3.

Enter desired revision name/number followed by **<Enter>** again.

F4 (TITLE BLOCK CONFIG. MENU)

Use: To assign a **DRAWING DATE** to the drawing associated with this configuration file. The drawing date assigned to this variable is included in the SCHPOST header (title) and is also placed in the title block of the applicable drawing.

Operation: When selected the **F4** variable lights. Press **<Enter>** and the following prompt is displayed:

Editing configuration parameter F4.

Enter applicable date (in any format) followed by **<Enter>** again.

F5 (TITLE BLOCK CONFIG. MENU)

Use: To assign the **MAXIMUM DRAWING PAGE NUMBER** to the drawing associated with this configuration file. The last page number assigned to this variable is placed in the title block of the applicable drawing.

Operation: When selected the **F5** variable lights. Press **<Enter>** and the following prompt is displayed:

Editing configuration parameter F5.

Enter the last page number of the drawing followed by **<Enter>** again.

F6, F7, F8 (TITLE BLOCK CONFIGURATION MENU)

Use: To assign user-defined variables to the drawing associated with this configuration file. As shipped, no SCHEMA III drawings/programs use the **F6**, **F7**, or **F8** variables. These variables may be assigned and used for any purpose.

Operation: Select the desired key. The applicable variable lights. Press **<Enter>** and the following prompt is displayed:

Editing configuration parameter Fx.

(where Fx is the applicable function key number.). Enter the desired value (up to 54 alphanumeric characters, characters above 54 will be truncated) followed by **<Enter>** again.

GRAPH (DISPLAY/EDIT CURRENT SETTINGS MENU)

Use: To edit the graphics adapter type. Applies to the SCHEMA.CFG file only.

Operation: When selected, the **GRAPH** command lights. Press **<Enter>** and the following prompt appears:

Editing configuration parameter GRAPH.

with the Text Line prompt allowing you to edit the current graphics adapter followed by **<Enter>** again.

If you are using a Sigma Designs Color 400 or Tecmar Graphics Master graphics adapter you can select a different screen color by entering a comma (,) and the applicable color number after the name of the graphics adapter. The color number is a single hex digit (**0-9** or **A-F**). For example:

,2 <Enter>

selects green as the screen color on a Sigma Designs Color 400 card.

IBM Enhanced Graphics Adapter (EGA) owners can select the desired

screen background color, cursor/image color, border color, drawing color, primary background, primary foreground, secondary background, secondary foreground. To do this, enter EG followed by four color fields, with each field separated by a comma. For example:

EG,8,37,3F,12,4,0,7,0

This entry selects the EGA adapter and sets the screen background color as dark purple (8), the cursor/image color as yellow (37), the border color as white (3F), the drawing color as green (12), the primary background as red (4), the primary foreground as black (0), the secondary background as grey (7), and the secondary foreground as black (0).

All EGA and VGA graphics drivers having at least 128K of RAM within the graphics board have four color fields: background, cursor/image, border and drawing. Each field can be one of 64 choices from 0 to 3F hex. The colors are determined by R, G, B, R', G' and B'. For example:

```
RG | B R' G' B'
0 1 | 0 0 1 0    BINARY (1 is "ON"  0 is "OFF")
1   |   2        HEX
```

12 is the color code for green. Experiment with the different color code combinations to find which colors you prefer.

See Also: Appendix E and SCHEMA III READ.ME file for the list of supported graphics adapters, and the command CONFIG-COLORS.

Notes: This variable is displayed when editing DRAWING.CFG files, but only the **GRAPH=** variable in the SCHEMA.CFG is actually used.

GRAPHICS ADAPTER SELECTION (HARDWARE CONFIG. MENU)

Use: To assign the graphics adapter type. Applies to the SCHEMA.CFG file only.

Operation: When selected, press **<Enter>** and a list of graphic adapters will appear with the following prompt:

Highlight the desired item and press <ENTER>.

After selecting the appropriate graphics adapter and pressing **<Enter>** the Hardware Configuration Menu will automatically appear.

When the desired graphics adapter has been entered, the Current Settings of the loaded configuration file are automatically updated. All changes should be **SAVE**d before exiting SCHCFG.

See Also: Appendix E and SCHEMA III READ.ME file for the list of supported graphics adapters.

Notes: This variable is displayed when editing DRAWING.CFG files, but only the **GRAPH=** variable in the SCHEMA.CFG is actually used.

INSERT (LIBRARIES CONFIGURATION MENU)

Use: To insert a blank object library entry in the Library Menu.

Operation: When selected the INSERT command lights. Press **<Enter>** and the INSERT command is activated. The INSERT command toggles ON and OFF. When activated, LIBR ON appears next to INSERT.

Select the library you want to insert followed by **<Enter>** again. A space is inserted at the specified LIBRx variable and any libraries assigned to higher LIBR variables are moved down.

For example, if LIBR7 is selected, press **<Enter>** and LIBR7 is cleared.

The library assigned to LIBR7 is now assigned to LIBR8 and the library assigned to LIBR8 is now assigned to LIBR9. If a library was assigned to LIBR9, it is lost (unassigned). Next, the following prompt is displayed:

Editing configuration parameter LIBRx.

where x is the library number to be inserted). Enter the full pathname with directory and extension for the new library.

See Also: DELETE command

Notes: Be sure to assign a library to the LIBRx variable you have inserted. When inserting a library and LIBR9 is selected, all information currently contained in LIBR9 will be overwritten. INSERT and DELETE are mutually exclusive commands.

LIBRx (LIBRARIES CONFIG. MENU)

Use: To assign an object library to **LIBRx** where x is 1-9.

Operation: When selected the **LIBRx** variable lights. Press **<Enter>** and the following prompt is displayed:

Editing configuration parameter LIBRx.

where x is 1-9. Enter the object library to be assigned to **LIBRx** followed by **<Enter>** again.

Notes: The full library name (including directory, name, and extension) must be entered with the **LIBRx** entry.

LOAD CONFIGURATION (MAIN CONFIGURATION MENU)

Use: To load an existing configuration file into SCHCFG.

Operation: When selected the **LOAD** command lights and the following prompt is displayed:

ENTER NAME OF .CFG FILE TO LOAD.

The cursor flashes next to the current directory path/file name, if applicable. If you wish to re-load the current file, press **<Enter>**.

Otherwise, press **<Esc>** and enter the desired directory path and configuration file name *with or without* the **.CFG** extension and press **<Enter>** again. For example:

\MEMORY\SCHEMA - Loads **SCHEMA.CFG** file from directory **MEMORY**

DRAWING - Loads **DRAWING.CFG** file from the current drawing directory

Notes: If you wish to see a list of all of the files with the **.CFG** extension residing in a directory, **\MEMORY**, enter the following:

\MEMORY*.CFG

Enter: **\MEMORY*.***

to see a list of all of the files.

MOUSE (DRAWING PARAMETERS CONFIGURATION MENU)

Use: To set the SCHDRAW mouse sensitivity. Applies to the SCHEMA.CFG file only.

Operation: When selected the **MOUSE** variable lights. Press **<Enter>** and the following prompt is displayed:

Editing configuration parameter MOUSE.

Enter the desired mouse horizontal sensitivity followed by a comma, vertical sensitivity, and **<Enter>** again. For example:

2,4 <Enter>

Notes: This configuration variable sets the ratio between mouse and cursor movement. This ratio is sometimes referred to as the mouse 'speed'. Values assigned to this variable are directly proportional to mouse speed, i.e., small values provide a small amount of mouse movement required to move the cursor a given distance.

Most users prefer a mouse sensitivity of **1,2**.

This variable is not displayed when editing DRAWING.CFG files.

PAN (DRAWING PARAMETERS CONFIGURATION MENU)

Use: To set the SCHDRAW load-time **CONFIG-PAN** Drawing Window panning speed.

Operation: When selected the **PAN** will toggle between the values of either **1, 2,** or **3. 2** is the default speed. **3** is a faster panning speed and is prefered by users that have high resolution graphics modes like 1024 x 768. The value of **1** may be used with lower resolution graphics like CGA.

See Also: Chapter 3, *SCHDRAW*

Notes: The value assigned to **PAN** can be changed from SCHDRAW using **CONFIG-PAN** and can be saved to the configuration file using **CONFIG-SAVE.**

PENW (DISPLAY/EDIT CURRENT SETTINGS MENU)

Use: To specify the width of the pen (only required for pen plotters).

Operation: When selected the **PENW** variable lights. Press **<Enter>** and the following prompt is displayed:

Editing configuration parameter PENW.

Enter the value as an integer where 1000 mils = 1 inch. If the width of your pen is given in millimeters, then divide the value by .0254 and truncate the result to an integer. For example, if the pen width is .35 mm, then the size in mils is .35/.0254 which equals 13.779. After truncation, the integer to be entered is:

13 <Enter>

See Also: Chapter 4, *SCHPLOT/SCHPRINT*

PLOTC (DISPLAY/EDIT CURRENT SETTINGS MENU)

Use: To assign plotter communications parameters. Applies to the SCHEMA.CFG file only.

Operation: When selected the PLOTC command lights. Press **<Enter>** and the following prompt is displayed:

Editing configuration parameter PLOTC.

with the Text Line prompt allowing you to edit the current communications port.

Enter the communications port number to which the plotter is connected. Then, press **<Enter>** again.

See Also: Chapter 4, *SCHPLOT/SCHPRINT*

PLOTTER (DISPLAY/EDIT CURRENT SETTINGS MENU)

Use: To assign the plotter type and optionally define a plotter initialization string.

Operation: When selected, the **PLOTTER** command lights. Press **<Enter>** and the following prompt appears:

Editing configuration parameter PLOTTER.

with the Text Line prompt allowing you to edit the current plotter type followed by **<Enter>** again.

Optionally, a comma and character string can be entered after the plotter type. The character string is sent to the plotter at the initialization time (beginning) of each plotted page. For example:

HI,VO <Enter>

selects Houston Instrument as the plotter type and outputs the characters **VO** to the plotter (sets the plotter pen velocity). Refer to the applicable plotter user's manual for information on initialization strings.

See Also: Appendix E and SCHEMA III READ.ME file for the list of supported plotters.

Notes: The communication port and baud rate are set through the **PLOTC** command when using the **DISPLAY/EDIT CURRENT SETTINGS** commands.

PLOTTER OPTIONS SELECTION (HARDWARE CONFIG. MENU)

Use: To assign the plotter type. Applies to the SCHEMA.CFG file only.

Operation: When selected the **PLOTTER** variable lights and the following prompt is displayed:

Highlight the desired item and press <ENTER>.

with a list of plotter types on the screen. Select the appropriate plotter and press **<Enter>**. The Hardware Configuration Menu automatically returns to the screen.

After the desired plotter configuration has been entered, the Current Settings for the loaded configuration file are automatically updated. All changes should be SAVEd before exiting SCHCFG.

See Also: Appendix E and SCHEMA III READ.ME file for the list of supported plotters.

PRINT (DISPLAY/EDIT CURRENT SETTINGS MENU

Use: To assign the printer type. Applies to the SCHEMA.CFG file ***only.***

Operation: When selected, the PRINT command lights. Press **<Enter>** and the following prompt will appear:

Editing configuration parameter PRINT.

with the Text Line prompt allowing you to edit the current printer driver setting. Be sure to enter the full path name of the printer driver and press **<Enter>** again. For example:

\DRV\DRIVER.DRV <Enter>

See Also: Appendix E and SCHEMA III READ.ME file for the list of supported printers.

Notes: This variable is displayed when editing DRAWING.CFG files, but only **PRINT=** variable in the SCHEMA.CFG is actually used.

PRINTER DRIVER SELECTION (HARDWARE CONFIG. MENU)

Use: To assign the printer type. Applies to the SCHEMA.CFG file only.

Operation: When selected, a screen will appear with the following prompt:

Highlight the desired item and press <ENTER>.

and a list of printer drivers will appear. After the desired printer driver

has been entered, the Current Settings of the loaded configuration file are automatically updated. All changes should be SAVEd before editing SCHCFG.

See Also: Appendix E and SCHEMA III READ.ME file for the list of supported printers.

Notes: This variable is displayed when editing DRAWING.CFG files, but only the PRINT= variable in the SCHEMA.CFG is actually used.

PROMPT (LIBRARIES CONFIGURATION MENU)

Use: To define the prompt field for floppy disk systems. The **PROMPT** variable is null for hard disk systems.

Operation: When selected the **PROMPT** variable lights. Press **<Enter>** and the following prompt is displayed:

Editing configuration parameter PROMPT.

Enter the desired prompt (any combination of 54 alphanumeric characters) followed by **<Enter>** again. For example:

PLACE LIBRARY DISK IN DRIVE A: AND PRESS ENTER

This prompt is generally used on floppy disk systems to allow time to replace the Program disk with the Standard Library disk.

Notes: This variable is displayed when editing DRAWING.CFG files, but only the **PROMPT=** variable in the SCHEMA.CFG is actually used.

PX(Y)LEN (DISPLAY/EDIT CURRENT SETTINGS MENU)

Use: To specify the length of the plot area in the x- and y-direction.

Operation: When selected the **PX(Y)LEN** variable lights. Press **<Enter>** and the following prompt is displayed:

Editing configuration parameter PX(Y)LEN

Enter the value as an integer where 1000 mils = 1 inch. For example, for **PXLEN**, if the plot size is 20.4 inches in the x-direction, this would equal 20400 mils. This integer would be entered:

20400 <Enter>

For **PYLEN**, if the plot size is 15.4 inches in the y-direction, this would equal 15400 mils. This integer would be entered:

15400 <Enter>

See Also: Chapter 4, *SCHPLOT/SCHPRINT*

Notes: This number is the plot size, not the paper size. Refer to your plotter manual for the maximum plot sizes for the paper you are using.

PX(Y)OFF (DISPLAY/EDIT CURRENT SETTINGS MENU)

Use: To specify the offset from the origin in the x- and y-direction.

Operation: When selected the **PX(Y)OFF** variable lights. Press **<Enter>** and the following prompt is displayed:

Editing configuration parameter PX(Y)OFF

Enter the value as an integer where 1000 mils = 1 inch. For example, for **PXOFF**, if the plot is to be shifted to the left of the origin 10.2 inches, this equals 10200 mils. This integer is entered:

-10200 <Enter>

For **PYOFF**, if the plot is to be shifted down from the physical origin 7.7 inches, this equals 7700 mils. The integer is entered:

-7700 <Enter>

See Also: Chapter 4, *SCHPLOT/SCHPRINT*

Notes: The origin for the larger HP plotters is the center of the physical page. In order to plot on the entire page, this parameter must be specified and must be negative. Refer to your plotter manual to determine the location of the origin on the physical page.

QUIT (MAIN CONFIG. MENU)

Use: To return to the Manager or DOS.

Operation: When selected the QUIT command lights. Be sure to save your file, if desired, before proceeding. Press **<Enter>** to return to the Manager or DOS.

Notes: Unsaved configuration information is lost when QUIT is selected. Be sure to SAVE the configuration information is so desired.

RPTX(Y) (DRAWING PARAMETERS CONFIGURATION MENU)

Use: To assign the SCHDRAW load-time **DRAW-REPEAT** horizontal (X) or vertical (Y) offset.

Operation: When selected the **RPTX** or **RPTY** variable lights. Press **<Enter>** and the following prompt is displayed:

Editing configuration parameter RPTX.

or

Editing configuration parameter RPTY.

Enter a numeric value corresponding to the number of pixels desired

between images repeated in SCHDRAW using the **DRAW-REPEAT** commands. Positive offset values place repeated images to the right of the original image. Negative values place repeated images to the left of the original image. After desired offset has been entered, press **<Enter>** again.

See Also: Chapter 3, *SCHDRAW*

Notes: The value assigned to **RPTX** or **RPTY** can be changed from SCHDRAW using **CONFIG-RPT.**

SAVE CONFIGURATION (MAIN CONFIGURATION MENU)

Use: To save the current, on-screen variables as a configuration file.

Operation: When selected the **SAVE** command lights. Press **<Enter>** and the cursor flashes next to the active directory path/file name, if applicable.

If you wish to overwrite the active file, press **<Enter>**. Otherwise, edit or erase by pressing **<Esc>** and enter the desired directory path and configuration file name *without* the **.CFG** extension and press **<Enter>** again. For example:

\MEMORY\SCHEMA <Enter> - Saves configuration file as **SCHEMA.CFG** in directory **MEMORY**

DRAWING <Enter> - Saves configuration file as **DRAWING.CFG** in the current directory.

Notes: Press **<Esc>** twice to return to the Main Configuration Menu and the configuration will not be **SAVE**d.

VGRID (DRAWING PARAMETERS CONFIGURATION MENU)

Use: To display or remove the SCHDRAW Visual Grid (**V GRID**) at load-time.

Operation: When selected the **VGRID** variable toggles between **ON** and **OFF**.

See Also: Chapter 3, *SCHDRAW*

Notes: **VGRID** is not associated with the cursor grid (**CGRID**).

The value assigned to **VGRID** can be changed from SCHDRAW using **CONFIG-V GRID**.

VLABEL (DRAWING PARAMETERS CONFIGURATION MENU)

Use: To activate the vertical icons for **DRAW-LABEL.**

Operation: When selected the **VLABEL** variable lights. Position the cursor over the variable and press **<Enter>** or the mouse and the variable will toggle between **OFF** and **ON.**

See Also: Chapter 3, SCHDRAW

Notes: If you normally do not use vertical labels then turn this variable off in the SCHEMA and DRAWING Configuration files.

WIHDR (DRAWING PARAMETERS CONFIGURATION MENU)

Use: To enable or disable SCHDRAW **CONFIG-WI HDR** (Wire Headers) at load-time.

Operation: When selected the **WIHDR** variable will toggle between **ON** and **OFF.**

See Also: Chapter 3, *SCHDRAW*

Notes: The value assigned to **WIHDR** can be changed from SCHDRAW using **CONFIG-WI HDR.**

XY (DRAWING PARAMETERS CONFIGURATION MENU)

Use: To enable or disable the SCHDRAW Cursor Coordinate Window at load-time.

Operation: When selected the XY variable lights and toggles between ON and OFF.

See Also: Chapter 3, *SCHDRAW*

ZOOM (DRAWING PARAMETERS CONFIGURATION MENU)

Use: To assign the default ZOOM level used by SCHDRAW.

Operation: When selected, the variable lights. Press **<Enter>**, and the following prompt is displayed:

Editing configuration parameter ZOOM.

Enter the desired ZOOM level at the Text Line prompt as follows:

Entry	Level
1	1:1
2	2:1
4	4:1
8	8:1

After a level is entered, press **<Enter>** again.

See Also: Chapter 3, *SCHDRAW*

Notes: If there is no ZOOM parameter specified and SCHDRAW has not been used to load a drawing, ZOOM will default to 1:1. Otherwise, the ZOOM level will be the same as the previous drawing.

Chapter 6. SCHLIB

6.1 INTRODUCTION

SCHLIB lets you create SCHEMA III object libraries, compress or enlarge libraries, move objects from one object library to another, and display object library status. SCHLIB also allows individual object manipulation within object libraries. Objects can be erased and moved between libraries. In addition, SCHLIB allows for quick Name Object creation/editing.

6.2 OBJECT LIBRARY COMPOSITION

Each SCHEMA III object library contains from one to over 4000 Name, Nested, and/or Body Objects. Even though SCHEMA III object libraries may contain thousands of objects, the libraries appear as single DOS files. Only SCHEMA III programs have the ability to read and write these object library files at the object level.

Each SCHEMA III object library consists of a directory and quantity of objects. When creating empty object libraries, the maximum number of possible objects the new library can hold is estimated. Object libraries that can hold 282, 1170, and 4640 objects can be created. Because the library's size is estimated in the beginning, adding new objects makes it necessary to enlarge the library's size to make room for more objects. This is done by creating a new, larger object library from an old, smaller object library. This simple procedure is described in paragraph 6.4.4.

Once a new, empty object library is created, objects can be added by

either copying them from existing object libraries or by creating new objects. New Name, Body, and Nested Objects are created in SCHDRAW's Object Editor, but Name Objects (only) can also be created using SCHLIB.

As objects are added to an object library, the library file size grows by each individual object's length. When an object is erased from an object library, the size of the library file does not decrease. When an updated version of an object is placed in an object library (the object already exists in the library), the original object stays in the library and the library's file size grows by the length of the new object.

Both of the above examples put extra, unnecessary entries in a library. These unnecessary entries can be removed by compressing the library. Paragraph 6.4.4 explains this operation in detail. SCHLIB provides the capability to display the number of bytes of wasted object space and the number of wasted directory entries that an object library currently has. Therefore, you can easily determine if an object library needs to be compressed.

6.3 INVOKING SCHLIB

Like most SCHEMA III programs, SCHLIB can be invoked from the Manager or directly from the DOS prompt. Chapter 2, *SCHEMA Program Manager*, contains detailed information on the Manager and the procedure to invoke SCHLIB. Whether you use the Manager or DOS, the SCHLIB command line is:

```
SCHLIB
```

Since SCHLIB operates on object library files, it does not use

configuration or drawing files as input. After **<Enter>** or the mouse button is pressed SCHLIB is invoked.

6.4 RUNNING SCHLIB

SCHLIB operation can be divided into three categories:

a. Library operations
b. General object operations
c. Name Object editing.

The first two operations are performed from the SCHLIB Main Menu, while the third operation is performed from the SCHLIB Name Object Editor Menu.

Library operations perform work on libraries as a whole, while general object and Name Object editing operations perform work on individual objects.

When SCHLIB loads, the screen is cleared and the SCHLIB Main Menu, shown in figure 6-1, is displayed. On floppy disk systems (after the Main Menu is displayed), the Program disk can be removed and replaced with the desired library disk.

The Main Menu display area displays library directories, DOS directories, and other data accessed through SCHLIB.

The library assignment line displays the libraries currently assigned to LIB1 and LIB2.

NOTE

You must assign an object library to LIB1 before using SCHLIB. There is no relationship between LIB1 and LIB2 in SCHLIB and the LIBR1 -

LIBR9 variables in the SCHEMA.CFG and DRAWING.CFG configuration files.

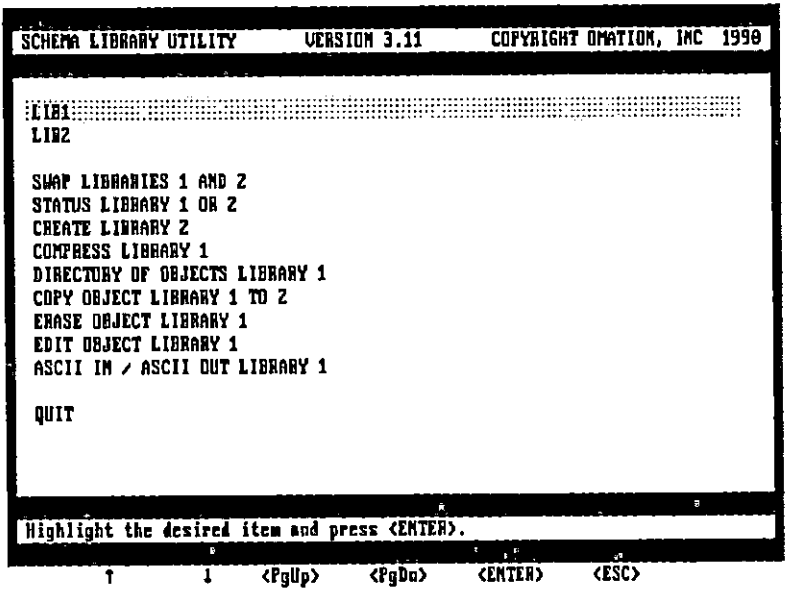

SCHEMA LIBRARY UTILITY VERSION 3.11 COPYRIGHT OMATION, INC 1990
LIB1
LIB2
SWAP LIBRARIES 1 AND 2
STATUS LIBRARY 1 OR 2
CREATE LIBRARY 2
COMPRESS LIBRARY 1
DIRECTORY OF OBJECTS LIBRARY 1
COPY OBJECT LIBRARY 1 TO 2
ERASE OBJECT LIBRARY 1
EDIT OBJECT LIBRARY 1
ASCII IN / ASCII OUT LIBRARY 1
QUIT
Highlight the desired item and press <ENTER>.
↑ ↓ <PgUp> <PgDn> <ENTER> <ESC>

Figure 6-1. SCHLIB Main Menu Format

The command columns list available SCHLIB commands. To select a command, highlight the command using the mouse button or the keyboard cursor keys and press **<Enter>**. A brief description of each command is contained in paragraphs 6.4.1 and 6.4.2. Paragraph 6.5 contains detailed descriptions of each SCHLIB command.

The prompt lines display SCHLIB user prompts.

The text line displays user keyboard entries. (Appendix C covers SCHEMA III text editing keyboard usage.).

When using SCHLIB's Name Object Editor, the Name Object Editor Menu shown in figure 6-2 is displayed.

The object text fields contain Name Object Description 1 through Description 4 fields (**NAME-DESC 1 - 4**) and the Name Object Body field (**NAME-BODY**).

The library assignment line displays the libraries currently assigned to LIBR 1 and LIBR 2.

The command columns list available SCHLIB Name Object Editor commands. To select a command, highlight the command using the mouse and pressing the left mouse button or keyboard cursor keys and press **<Enter>**. The right mouse button is the same as pressing the **<Esc>** key. A brief description of each SCHLIB Name Object Editor command is contained in paragraph 6.4.3. Detailed descriptions of the commands are contained in paragraph 6.5.

The prompt lines display SCHLIB Name Object Editor user prompts.

The text line displays user keyboard entries.

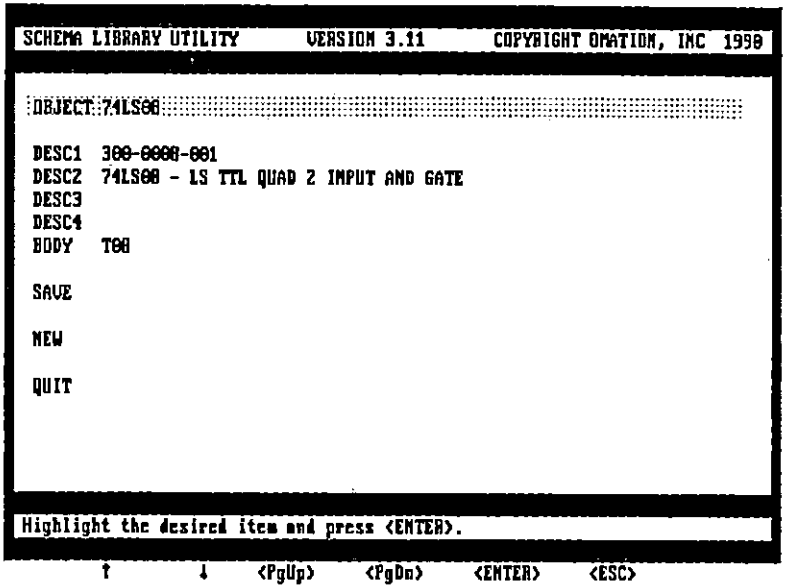


Figure 6-2. SCHLIB Name Object Editor Menu Format

6.4.1 Main Menu Library Operations

Before you can use SCHLIB, you need to specify the library(s) you want to use. Only two libraries can be worked with at once. These libraries are called LIB1 and LIB2 (not to be confused with the LIBR1 - LIBR9 variables in the SCHEMA III configuration files). Libraries are assigned to LIB1 and LIB2 using the LIB1 and LIB2 commands. The remaining Main Menu library operation commands are:

SWAP LIBRARIES 1 AND 2 - Used to exchange libraries assigned to LIB1 and LIB2. This is most useful when moving objects between the two libraries.

STATUS LIBRARY 1 OR 2 - Used to display library status.

CREATE LIBRARY 2 - Used to create a new, empty library. Also used to estimate the library's size.

COMPRESS LIBRARY 1 - Used to compress or expand a library.

6.4.2 Main Menu General Object Operations

The Main Menu general object operations pertain to all SCHEMA III Name, Nested, and Body Objects. The object operation commands are:

DIRECTORY OF OBJECTS LIBRARY 1 - Used to display an alphabetized list of all, or a selected portion of, a given library's directory. The output can be redirected to a printer, if desired.

COPY OBJECT LIBRARY 1 TO 2 - Used to copy a single object, groups of objects, or all objects from the library assigned to LIB1 to the library assigned to LIB2.

ERASE OBJECT LIBRARY 1 - Used to erase single objects or groups of objects from the library assigned to LIB1.

ASCII IN / ASCII OUT LIBRARY 1 - Used to ASCII IN or Name Objects from a file, or ASCII OUT Name Objects to a file for the library assigned to LIB1.

6.4.3 Name Object Editor Menu

SCHLIB's Name Object Editor Menu is entered through the **EDIT OBJECT LIBRARY 1** command from the Main Menu. When **EDIT**

OBJECT LIBRARY 1 is invoked, the Main Menu is replaced with the Name Object Editor Menu. The Name Object Editor is used to edit or create Name Objects. The Name Object Editor commands are:

OBJECT - Used to load an existing Name Object (from the library assigned to LIB1) into the editor for editing

DESC1-DESC4, BODY - Used to create/edit the applicable fields of the current Name Object.

SAVE - Used to save the current Name Object to the library assigned to LIB1

NEW - Used to initialize all entries to empty

QUIT - To return to the SCHLIB Main Menu.

6.4.4 Compressing Libraries and Expanding Directories

To compress a library or expand a library's directory size, a new library must be created with the contents of the original library. After the new library is completed, it is renamed to the original library's name. There is a manual alternative for using the **COMPRESS LIBRARY 1** command in SCHLIB. The following paragraphs describe this process:

a. Create an empty library of the desired size using the **CREATE LIB** command. This is the library that will be renamed to the name of the library being compressed/enlarged. If you are compressing a library, make the new library the same size as the current library. If you are enlarging a library directory size, make the new library larger than the current library. The **CREATE LIB** command automatically assigns this new library to LIB2.

b. Use the **SET LIB** command to assign the library you wish to compress/expand to LIB1.

c. Use the **COPY OBJ** command to copy all objects from LIB1 to LIB2 via a * object name entry. This operation can take a few minutes if a large number of objects are copied.

d. Exit SCHLIB.

e. From DOS, either erase or rename the original library.

f. Rename the new library to the old library's name.

6.4.5 Library Philosophy

We suggest that you do not add any new parts to our nine libraries. You should create a User library to store any new parts that are created. This library would be the only library necessary to backup in case of hard disk failure.

6.4.6 Assign a User Library

After a user library is created it can be assigned to the default position, LIBR1 in the DRAWING.CFG. Where drawing represents the name of the schematic. Thus, when using the Object Editor in SCHDRAW and you save a part and do not specifiy any particular library, it will automatically be saved to your library in the #1 position. Use the SCHCFG program to place your new library into the LIBR1 position of the DRAWING.CFG. Also you may wish to place the new library into the LIBR1 position of the SCHEMA.CFG file, since this is the template file from which all DRAWING.CFG files are created. Then all new projects begun will have your new library in LIBR1

6.4.7 Archive a Project

Let's say that you have just created a ten page schematic project, PROJECTX.P01 through PROJECTX.P10. However, now you wish to send it to a collegue or store it away , but you don't want to save away all the libraries with it. This is when you should localize your libraries using the program, SCHLOCAL. Use SCHLIB to create a library called PROJECTX.LBR and place this library into the **LIBR1** position of the PROJECT.CFG. Then execute the SCHLOCAL on PROJECTX. Thus all the parts used in the schematic are copied to the first library, PROJECTX.LBR. Libraries 2 - 9 can be nullified and only the first library will be needed.

6.5 COMMAND DESCRIPTIONS

This paragraph contains detailed descriptions of SCHLIB commands. Each command is described as follows:

- **Use:** Defines reason for use

- **Operation:** Lists procedure to use command

- **See Also:** Identifies related commands/information, if applicable

- **Notes:** Identifies special conditions, limitations, etc. associated with command, if applicable.

ASCII IN / ASCII OUT LIBRARY 1

Use: To Ascii In or Out Name Objects to or from a file.

Operation: When selected, the ASCII IN / ASCII OUT command lights. Press **<Enter>** and a new menu is displayed:

LIB 1 -> C:\SCHEMA\XXXXXXXX.LBR

ASCII IN NAME OBJECTS TO LIB 1
ASCII IN SYNTAX CHECK

ASCII OUT NAME OBJECTS FROM LIB 1

QUIT TO MAIN MENU

Select the desired command and press **<Enter>**.

ASCII IN NAME OBJECTS TO LIB 1 (ASCII IN/OUT MENU)

Use: To ASCII IN Name Objects into the library assigned to LIB 1.

Operation: First you would have a ASCII file that was created or altered with a non-document text editor. The format is as follows:

```
[
NAME
DESC 1
DESC 2
DESC 3
DESC 4
BODY
]
```

All items must be present if a DESC field is not to be used then leave a blank space in its place. When this command is selected the prompt is displayed:

ENTER FILENAME TO ASCII IN -> LIB1

Enter the name of the file you wish to ASCII IN. The syntax is checked as it is read. If there are any errors a file with the same prefix as the input file with a .ERR extension will be created. The block of input lines will be sent to the file with error messages. This error file can be altered and re-read into the library. The error messages will be ignored since they occur between brackets.

ASCII IN SYNTAX CHECK (ASCII IN/OUT MENU)

Use: To check the Ascii syntax before trying to Ascii in Name Objects into the library assigned to LIB 1.

Operation: When selected the prompt at the bottom of the menu says:

ENTER FILENAME TO CHECK ASCII IN SYNTAX

Type in the path and filename of the file to check the syntax.

ASCII OUT NAME OBJECTS LIB 1 (ASCII IN/OUT MENU)

Use: To Ascii out Name Objects from the library assigned to LIB 1.

Operation: When selected the prompt at the bottom of the menu says:

ENTER FILENAME TO ASCII OUT FROM LIB 1

Type in the path and filename of the file to Ascii out to.

Then the prompt says:

ENTER NAME OBJECTS TO ASCII OUT -> FILENAME
PRESS ESC TO QUIT

Enter the Name Objects that you wish to transfer to the file. Wildcards can be used.

See Also: ASCII IN NAME OBJECTS TO LIB 1
for information on the proper syntax for the ASCII format.

BODY (NAME OBJ. ED. MENU)

Use: To create/edit Name Object **NAME-BODY** fields.

Operation: When selected, the **BODY** command lights. Press **<Enter>** and the following prompt is displayed:

ENTER/MODIFY BODY TEXT

The current **BODY** field (if applicable) appears in the text line and can be edited. If no field is present, a blank line allows text to be entered. Enter desired **BODY** text followed by **<Enter>**. The entered text appears by the **BODY** field in the display area.

Notes: The **BODY** field contains the prefix of the associated Body Object name.

COMPRESS LIBRARY 1 (MAIN MENU)

Use: To compress a library or expand a library's directory size.

Operation: When selected, the **COMPRESS** command lights. Press **<Enter>** and the prompt appears:

ENTER SIZE OF LIBRARY TO BE CREATED - S, M, L

Enter the appropriate size and press **<Enter>**. The library is then compressed and the SCHLIB Main Menu appears.

See Also: Section 6.4.4, *Compressing Libraries and Expanding Directories*

Notes: When a library is compressed, a backup file is created with the **.LBK** extension.

COPY OBJECT LIBRARY 1 TO 2 (MAIN MENU)

Use: To copy a Name, Nested, or Body Object from the library assigned to LIB1 to the library assigned to LIB2.

Operation: When the **COPY OBJECT LIBRARY 1 TO 2** command lights. Press **<Enter>** and the following prompt is displayed:

ENTER SOURCE (IN LIBR 1) THEN DESTINATION (IN LIBR 2)
OR PRESS ESC TO QUIT

Enter the name of the object you would like to copy from the library assigned to LIB1 to the library assigned to LIB2 followed by a **<Space>** and its new name (if applicable). There are four types of copies:

a. Copying and renaming a single object
b. Copying a single object with the same name
c. Copying a group of objects
d. Copying all objects.

Examples of the four types are as follows:

a. **74LS00<Space>74LS08** (Copies the object **74LS00** from LIB1 to LIB2 renaming it as **74LS08**)

b. **74LS00 <Enter>** (Copies **74LS00** from LIB1 to LIB2 as a **74LS00**)

c. **74LS* <Enter>** (Copies all **74LS** objects from LIB1 to LIB2 without renaming the objects)

d. *** <Enter>** (Copies all objects from LIB1 to LIB2 without renaming the objects)

After the copy is complete, the Main Menu displays:

n OBJECTS COPIED
Press any key to continue ...

(where: **n** is the number of objects copied.).

After **<Enter>** is pressed, the original prompt is displayed so additional objects can be copied:

ENTER SOURCE (IN LIB1) THEN DESTINATION (IN LIB2)

Press **<Esc>** to return to the SCHLIB Main Menu.

Notes: Wildcard (*) or group copies can not be renamed. For example:

74LS*<Space>54LS* <Enter>

is an illegal copy entry.

When using a wildcard (*) to do multiple or group object copies, a message will appear as follows:

OVERWRITE IF OBJECT EXISTS? (Y/N)

If Y is selected, the following prompt will appear:

PROMPT BEFORE OVERWRITE? (Y/N)

If Y is selected, every time an object from the source library is instructed to overwrite an object in the destination library the following message appears:

OBJECT XXXX ALREADY EXISTS, OVERWRITE? (Y/N)

If Y is selected, the object from the source library will overwrite the object in the destination library. Otherwise, if N is selected, the object in the source library will not overwrite the object in the destination library.

CREATE LIBRARY 2 (MAIN MENU)

Use: To create a new, empty object library.

Operation: When selected, the **CREATE LIBRARY 2** command lights. Press **<Enter>** and the following prompt is displayed:

ENTER SIZE OF LIBRARY TO BE CREATED - S, M, OR L

Enter desired library size (S - small libraries; can contain up to 282 directory entries (objects), M - medium libraries; can contain up to 1170 entries, and L - large libraries; can contain up to approximately 4,640 entries) followed by **<Enter>**. The following prompt is displayed:

ENTER LIBRARY NAME WITH DIRECTORY AND EXTENSION (.LBR)

Enter directory (if applicable) and library name (with .LBR extension) followed by **<Enter>**. A new DOS file is created in the specified directory with the assigned library name. If no directory is specified, the library is created in the current directory. The new library is automatically assigned to LIB2. Upon completion, the Main Menu appears.

Notes: Empty, small libraries occupy approximately 4K bytes of disk space. Empty, medium libraries occupy approximately 16K bytes. Large, empty libraries occupy approximately 64K bytes. As objects are added to a new library, the library size grows by the length of each object.

If a file already exists with the same name as the created library, the original file is erased.

DESC1 (NAME OBJ. ED. MENU)

Use: To create/edit Name Object Description 1 (NAME-DESC 1) fields.

Operation: When selected, the DESC1 command lights. Press **<Enter>** and the following prompt is displayed:

ENTER/MODIFY DESC TEXT

The current **DESC1** field (if applicable) appears in the text line and can be edited. If no field is present, a blank line allows text to be entered. Enter desired **DESC1** text followed by **<Enter>**. The entered text appears by the **DESC1** field in the display area.

Notes: Name Objects shipped with SCHEMA III predefine the DESC1 field as the in-house (company) part number for the SCHPOST post-processors.

DESC2 (NAME OBJ. ED. MENU)

Use: To create/edit Name Object Description 2 (NAME-DESC 2) fields.

Operation: When selected, the **DESC2** command lights. Press **<Enter>** and the following prompt is displayed:

ENTER/MODIFY DESC TEXT

The current **DESC2** field (if applicable) appears in the text line and can be edited. If no field is present, a blank line allows text to be entered. Enter desired **DESC2** text followed by **<Enter>**. The entered text appears by the **DESC2** field in the display area.

Notes: Name Objects shipped with SCHEMA III predefine the DESC 2 field as the component description for the SCHPOST post-processors. For example: **74LS00 - QUAD 2-INPUT NAND GATE**

DESC3 (NAME OBJ. ED. MENU)

Use: To create/edit Name Object Description 3 (**NAME-DESC 3**) fields.

Operation: When selected, the **DESC3** command lights. Press **<Enter>** and the following prompt is displayed:

ENTER/MODIFY DESC TEXT

The current **DESC3** field (if applicable) appears in the text line and can be edited. If no field is present, a blank line allows text to be entered. Enter desired **DESC3** text followed by **<Enter>**. The entered text appears by the **DESC3** field in the display area.

Notes: The **DESC3** field is user-defined.

DESC4 (NAME OBJ. ED. MENU)

Use: To create/edit Name Object Description 4 (**NAME-DESC 4**) fields.

Operation: When selected, the **DESC4** command lights. Press **<Enter>** and the following prompt is displayed:

ENTER/MODIFY DESC TEXT

The current **DESC4** field (if applicable) appears in the text line and can be edited. If no field is present, a blank line allows text to be entered. Enter desired **DESC4** text followed by **<Enter>**. The entered text

appears by the DESC4 field in the display area.

Notes: The DESC4 field is user-defined.

DIRECTORY OF OBJECTS LIB. 1 (MAIN MENU)

Use: To display or print a library directory, i.e., the names of the objects in the library.

Operation: When selected, the DIRECTORY OF OBJECTS LIBRARY 1 command lights. Press <Enter> and the following prompt is displayed:

ENTER SEARCH STRING

The search string format is:

ABCD <Enter> (Displays object named ABCD)

ABC* <Enter> (Displays all objects beginning with ABC)

* <Enter> (Displays all objects in the library).

PRINT LIBRARIES

If <Space>\ is entered after the search string, the output is redirected to your printer.

After **<Enter>** is pressed, matching object names are displayed in the display area.

If there are multiple screens used to display the objects in a particular library, use the **<Arrow>, <PgDn>, <PgUp>, <Home>** and **<End>** keys.

Examples:

74LS* <Enter> (Displays all objects in the library starting with 74LS)

*** <Enter>** (Displays all objects in the library)

74LS*<Space>\ <Enter> (Prints all objects in the library starting with 74LS).

Notes: Select QUIT to return to the SCHLIB Main Menu.

Library directories are displayed alphanumerically.

EDIT OBJECT LIBRARY 1 (MAIN MENU)

Use: To access the Name Object Editor to create/edit Name Objects.

Operation: When selected, the Main Menu is replaced with the Name Object Editor Menu. The following options appear on the screen:

OBJECT
DESC1
DESC2
DESC3
BODY
SAVE
NEW

QUIT

Notes: Name Object Editor Menu commands only apply to LIB1. Thus, the desired object library must be assigned to LIB1 before executing **EDIT OBJECT LIBRARY 1.**

ERASE OBJECT LIBRARY 1 (MAIN MENU)

Use: To erase a Name, Nested, or Body Object from the library assigned to LIBR 1.

Operation: When selected, the **ERASE OBJECT LIBRARY 1** command lights. Press **<Enter>** and the following prompt is displayed:

ENTER NAME OF OBJECT TO BE ERASED
(OBJECT IS IN LIBR 1) OR PRESS ESC TO QUIT

Enter the name of the object to be erased. Valid entries are as follows:

ABCD <Enter> (Erases object ABCD)

ABC* <Enter> (Erases all objects in LIB1 that begin with ABC)

*** <Enter>** (Erases all objects in LIB1).

After the object(s) is erased, the **ENTER NAME OF OBJECT TO BE ERASED (OBJECT IS IN LIBR 1)** prompt is again displayed.

Press **<Esc>** to return to the SCHLIB Main Menu.

See Also: Paragraph 6.4.4, Compressing Libraries and Expanding Directories

Notes: Erasing an object does not decrease library size - the object's name is simply removed from the library's directory. Compressing the library recovers this space (see paragraph 6.4.4).

LIB1 (MAIN MENU)

Use: To assign libraries to LIB1 for use with all other SCHLIB commands. You must use this command before executing any other SCHLIB editing commands.

Operation: When selected, the **LIB1** command lights. Press **<Enter>** and the following prompt is displayed:

ENTER PATH AND FILE FILTER

Enter library path and name (including extension) followed by

<Enter>. If the library is located and opened, the library name is assigned and displayed in the library assignment line.

If the library is not located, an error message is displayed. The current entry should be edited to reflect the valid library path and name.

Notes: If the desired library does not reside within the current directory, be sure to include the path before the library name, such as:

\LIBR\TTL.LBR <Enter>

Both valid DOS wildcard characters (* ?) are supported by DIR LIB. After the search string has been entered, press **<Enter>**. The specified DOS file(s) is displayed. For example:

\LIBR*.LBR <Enter> (Displays all files with the extension of .LBR in directory LIBR)

\LIBR\TTL?.LBR <Enter> (Displays all files that are four characters in length and have TTL as the first three characters with the .LBR extension in directory LIBR)

TTL.LBR <Enter> (Displays file TTL.LBR in the current directory).

Do not, under any circumstances, assign the same library to LIB1 and LIB2 as library contents may be lost during writes to the libraries.

LIB2 (MAIN MENU)

Use: To assign libraries to LIB2 for use with all other SCHLIB commands. You must use this command before executing any other SCHLIB editing commands.

Operation: When selected, the **LIB2** command lights. Press **<Enter>** and the following prompt is displayed:

ENTER PATH AND FILE FILTER

Enter library path and name (including extension) followed by **<Enter>**. If the library is located and opened, the library name is assigned and displayed in the library assignment line.

If the library is not located, an error message is displayed. The current entry should be edited to reflect the valid library path and name.

Notes: If the desired library does not reside within the current directory, be sure to include the path before the library name, such as:

\LIBR\TTL.LBR <Enter>

Both valid DOS wildcard characters (* ?) are supported by DIR LIB. After the search string has been entered, press **<Enter>**. The specified DOS file(s) is displayed. For example:

\LIBR*.LBR <Enter> (Displays all files with the extension of .LBR in directory LIBR)

\LIBR\TTL?.LBR <Enter> (Displays all files that are four characters in length and have TTL as the first three

characters with the .LBR extension in directory LIBR)

TTL.LBR <Enter> (Displays file TTL.LBR in the current directory).

Do not, under any circumstances, assign the same library to LIB1 and LIB2 as library contents may be lost during writes to the libraries.

NEW (NAME OBJ. ED. MENU)

Use: To clear the current Name Object DESC1 - 4 and BODY fields.

Operation: When selected, the NEW command flashes while the current Name Object DESC1 - DESC 4 and BODY fields in the display area are cleared.

OBJECT (NAME OBJ. ED. MENU)

Use: To load an existing Name Object from the library assigned to LIB1 into the Name Object Editor.

Operation: When selected, the OBJECT command lights. Press <Enter> and the following prompt is displayed:

ENTER SEARCH STRING

Enter desired Name Object name followed by **<Enter>**. If found, the Name Object is loaded and its **DESC1 - DESC4** and **BODY** fields are displayed in their respective display area locations. The fields can now be edited.

If a wildcard character (*) is used, the listing of objects appears on the screen. The desired object can then be loaded.

Notes: Trying to load Body or Nested Objects instead of Name Objects results in an error message.

QUIT (MAIN MENU)

Use: To return to the Manager or DOS.

Operation: When selected, the QUIT command lights.

Press **<Enter>** to return to the Manager or DOS.

Notes: The actual disk libraries are updated by this command before returning to the Manager or DOS. Thus, if DOS is accessed in another manner (for example, simultaneously pressing **<Ctrl><Break>**), all library updates are lost.

QUIT (NAME OBJ. ED. MENU)

Use: To return to the Main Menu.

Operation: When selected, the QUIT command lights. Press **<Enter>** to return to the SCHLIB Main Menu.

Notes: The current Name Object is cleared if QUIT is executed before saving. Thus, **SAVE** the current Name Object (if desired) before returning to the Main Menu.

SAVE (NAME OBJ. ED. MENU)

Use: To save the current Name Object to the library assigned to LIB1.

Operation: When selected, the SAVE command lights. Press **<Enter>** and the following prompt is displayed:

ENTER OBJECT NAME (SAVES TO LIBR1)

Enter name of Name Object to be saved followed by **<Enter>**.

Notes: If the object already exists the Object Editor Menu displays:

OBJECT ALREADY EXISTS, PRESS
ENTER TO SAVE, ANY OTHER KEY TO ABORT

Pressing **<Enter>** overwrites the existing object. If any other key is

pressed, the Object Editor Menu with the following prompt is displayed:

ENTER OBJECT NAME (SAVES TO LIBR1)

STATUS LIBRARY 1 OR 2 (MAIN MENU)

Use: To display library status.

Operation: When selected, the **STATUS LIBRARY 1 OR 2** command lights. Press **<Enter>** and the following prompt is displayed:

ENTER LIBRARY NUMBER, 1 OR 2

Enter applicable library number followed by **<Enter>**. The following status message is displayed:

LIBRARY: nnn
TOTAL LIBRARY SIZE BYTES: nnn
TOTAL RECOVERABLE BYTES: nnn
NUMBER OF IN USE DIRECTORY ENTRIES: nnn
NUMBER OF RECOVERABLE DIRECTORY ENTRIES: nnn
NUMBER OF FREE DIRECTORY ENTRIES: nnn

where:

LIBRARY: is the name of the selected library.
TOTAL LIBRARY SIZE BYTES: is the library's DOS file size in bytes.

TOTAL RECOVERABLE BYTES: is the number of bytes that can be recovered by library compression.
NUMBER OF IN USE DIRECTORY ENTRIES: is the number of valid objects in this library.
NUMBER OF RECOVERABLE DIRECTORY ENTRIES: is the number of directory entries that can be recovered by library compression.
NUMBER OF FREE DIRECTORY ENTRIES: is the number of available directory entries.

Press **<Esc>** to return to the SCHLIB Main Menu.

SWAP LIBRARIES 1 AND 2 (MAIN MENU)

Use: To exchange the LIB1 and LIB2 assignments.

Operation: When selected, the **SWAP LIBRARIES 1 AND 2** command briefly flashes while library assignments are exchanged.

The library that was assigned to LIB1 is now assigned to LIB2 and the library that was assigned to LIB2 is now assigned to LIB1.

Notes: **SWAP LIBRARIES 1 AND 2** only changes library *assignments* and has no affect on library contents.

Chapter 7. SCHPOST

7.1 INTRODUCTION

SCHPOST provides fourteen post-processing functions for SCHEMA III:

a. Routing
b. Bill of Materials
c. LOTUS Output File
d. DBASE Output File
e. DELIMITED Output File
f. SDF Output File
g. Design Rule Check
h. Net List
i. Pin List
j. Wire List
k. Reports
l. SCHEMA-PCB Pin List
m. Net/Pin List Conversion
n. Build All

SCHPOST extracts data from schematic drawing files (produced in SCHDRAW) to produce these post-processor outputs. A description of each post-processor is contained in the following paragraphs.

7.2 POST-PROCESSORS

The SCHPOST post-processors extract and analyze schematic file data. The Routing post-processor automatically executes when SCHPOST is invoked. Remaining post-processors must be selected before they can be used.

This SCHPOST in an incrementally compiling post-processor. Each top level page is compiled into its own object file that roughly contains a pinlist for that page. The file has the extension Oxx, where xx is the page number. Then all the pages are merged and sorted together to create two files. The .PST, which is a part list file and the .NST, which is the netlist file. These are the two files that all the outputs will use as inputs. If you later make a change to a page of the schematic, then that page will be the only one to be re-processed. If the .NST and .PST files are newer than the schematic pages and their object files, Oxx, then no processing will need to be repeated. The **BUILD ALL** function is there to force SCHPOST to process the whole schematic.

WARNING: SCHPOST will *NOT* be able to detect whether you have changed a library. If you have changed a library part that affects your schematic use, BUILD ALL

7.2.1 Routing

The Routing post-processor checks the schematic for the following basic drawing connection errors:

a. Signal wires and labels that are not connected to an object, label, or another signal wire
b. Connection dots that have an improper number of intersecting signal wires/objects
c. Reference Designators used two or more times
d. Nets with more than one label.

These errors are the lowest-level errors that can occur and generally point to a portion of the schematic drawing that has been drawn incorrectly. Get in the habit of running SCHPOST so the Routing post-processor will execute, even if none of the other post-processors are to be used. If no errors are located by the Routing post-processor, there is no Routing post-processor output.

The reference designator usage check searches for multiple uses of identical (redundant) reference designators, i.e., functionally different components with the same assigned reference designator. Identifying different *sections* (gates) of the same component (such as a quad AND gate) with the same reference designator is legal and thus, is not flagged as an error.

For example, the following Schematic Editor DRAW-OBJECT Parameter String entries are not flagged as an error:

74LS00\A,U1 and **74LS00\B,U1**

See paragraph 3.4.4.3 for information on suppressing this error for certain objects.

The nets that have more than one assigned label is an error that generally indicates that two unique nets have inadvertently been connected together or a net has been assigned two different labels.

7.2.2 Bill of Materials

The Bill of Materials (BOM) post-processor extracts part type, quantity, and description (DESC) data from schematics. To be included in the BOM output, each schematic component Body Object must, at a minimum, have the following defined:

a. A reference designator location (**BODY-PARM R**)
b. At least one pin definition (**BODY-DF PI**) or (**BODY-PIN**).

The actual output format of the BOM is user-defined and quite flexible. The BOM post-processor generates a separate output line for each component (or alternately, for each different component) used in the associated schematic.

Each output line can contain any of the following fields, arranged in any order and field width:

a. Item Number
b. Quantity
c. Part Type
d. Reference Designator
e. Name Object Descriptions (**DESC 1 - DESC 4**).

Optional title blocks and headings can be added to output lines. In addition, output lines can be sorted and combined. The versatility of the BOM post-processor allows most bill of material formats to easily be reproduced.

In addition to the standard text format, the Bill of Materials can also be output in LOTUS (**.WKS**), DBASE (**.DBF**), DELIMITED (**.TXT**) and SDF (**.SDF**) formats. These file outputs allow the user to perform a number of different spreadsheet and database operations on the Bill of

Materials data. Operations such as cost analysis and inventory record keeping are made easier through the use of these file formats.

7.2.3 Design Rule Check

The Design Rule Check post-processor checks for complex drawing errors. Five different types of errors are identified by the Design Rule Check post-processor:

a. Floating Inputs
b. Net Configuration (four checks).

The floating input check searches all component pins assigned as inputs (BODY-DF PI pins defined as I) for connections. If an input pin is found unconnected (floating), an error message is generated. If anything is connected to the applicable input pin, no error message is generated.

The first net analysis searches for input-only nets, i.e., nets that consist of pins defined as inputs with no other pin types present. If an input-only net is discovered, an error message is generated. If any other pin type is present in the net (regardless of type), an error message is not generated.

The second analysis searches for nets that contain more than one output, i.e., nets that have two or more pins defined as outputs. Any number and type of other pins may be present, but as long as two or more outputs exist, an error message is generated.

The third net analysis searches for nets with only one component, i.e., a pin is assigned a label but the label is not used anywhere else on the schematic (the net consists of a single pin).

The fourth net analysis searches for pins on a component that have two different net names (labels) assigned to it. The most common way that this can occur is if the user assignes a pin definition twice on the same part. Then this part has two different nets attached to these like pins.

7.2.4 Net List

The Net List post-processor generates a machine-readable net list file from the applicable input schematic file. The net list file identifies all nets on the applicable schematic and contains information on each component type. The file format is very simple - see Appendix B for a complete description of the net list file format.

7.2.5 Pin List

The Pin List post-processor is similar to the Net List post-processor but contains more information. While the net list contains connection information arranged by net (i.e., a list of components connected together), a pin list contains information arranged by device, i.e., the pin list identifies a device and its pins and net names. The pin list file format is very simple - see Appendix B for a complete description of the format. The SCHEMA-PCB pin list post-processor generates a SCHEMA-PCB-compatible pin list for input to SCHEMA-PCB.

7.2.6 Wire List

The Wire List post-processor generates a wire list from the input schematic file. The wire list contains the same connection information as the net list (complete set of nets for the schematic) except it is in a human-readable form. The wire list is used to verify proper net list

extraction and supplement general design documentation. The wire list also greatly aids hardware prototype construction.

7.2.7 Reports

The Reports post-processor generates reference designator and signal label usage reports from the input schematic file. These usage reports aid design verification and documentation preparation.

The reference designator usage report alphabetically lists all reference designators in the applicable schematic file. Accompanying each reference designator is a list of all its applicable sub-types (such as individual sections of a quad AND gate) and the schematic page number(s) the sub-types appear on.

The signal label usage report is an alphabetical listing of all signal labels. Accompanying each signal label listing is the schematic page number(s) on which the signal label appears.

7.3 OPERATION

7.3.1 Invoking SCHPOST

Like most SCHEMA III programs, SCHPOST can be invoked from the Manager or directly from the DOS prompt. Chapter 2, *SCHEMA Program Manager*, contains detailed information on the Manager and the procedure to invoke SCHPOST. The following message appears on the command line of the Manager screen:

SCHPOST FILENAME (LESS EXTENSION)

When you invoke from DOS, the SCHPOST command line should be entered in the following format:

SCHPOST DRAWING POST-PROCESSOR(S) OUTPUT

where **DRAWING** is the name of the schematic drawing (less page number extension) to process. **DRAWING** must specify the complete path and file name of the desired drawing.

The optional **POST-PROCESSOR(S)** field identifies which post-processing functions are desired. Valid entries include one or more of the following:

A (All post-processors except SCHEMA-PCB Pin List)

B (Bill of Materials)

D (Design Rule Check)

N (Net List)

P (Pin List)

C (Net/Pin List Conversion)

R (Reports)

L (LOTUS Output File)

G (DBASE Output File)

I (DELIMITED Output File)

J (SDF Output File)

S (SCHEMA-PCB Pin List)

W (Wire List).

U (BUILD ALL compiles the whole schematic)

Z (Supress the hierarchical prefix for Reference designators)

When multiple entries are used, do not separate entries with spaces. For example, BN selects the Bill of Materials (BOM) and Net List post-processors. The only way to run the **Routing Check** is to leave the **POST-PROCESSOR(S)** field blank. The SCHPOST post-processor options appear. At this time, you may select **Routing Check**. It can be invoked *only* from the selection screen.

The optional **OUTPUT** field is the standard DOS output redirection option. If the **OUTPUT** field is left blank, all output defaults to the console (screen). To redirect the output to a file, enter a greater-than sign (>) followed by the desired file name from the DOS command line. For example, to direct the output to your printer, enter:

>PRN <Enter>

Outputs from all post-processors (except the net and pin lists, Lotus, Dbase, Delimited and SDF) are routed to the selected device. The above exceptions are sent to files. The net list file is named after the associated schematic with a file extension of .NET, pin list - .NPL, Lotus - .WKS, Dbase - .DBF, Delimited - .TXT and SDF - .SDF. See your DOS user's manual for additional information on output redirection. Example SCHPOST command lines are as follows:

SCHPOST LOGIC A - Processes the schematic **LOGIC** from the current directory and runs all post-processors. All output is directed to the screen.

SCHPOST \SCHEMA\LOGIC UBDN >MEMORY.XXX - Processes the entire schematic **LOGIC** using Build All in the **SCHEMA** directory and only runs the Bill of Materials, Design Rule Check, and Net List post-processors. The output is directed to file **MEMORY.XXX**.

7.3.2 Running SCHPOST

After the SCHPOST command line from DOS is entered, the following prompt is displayed:

SCHEMA POST PROCESSOR Version X.XX
Copyright Omation, Inc 1989

(where: **X.XX** is the current version number.).

If the command line post-processor(s) field was left blank when SCHPOST was invoked, the following options appear:

ALL SUB POST-PROCESSORS
BILL OF MATERIALS
DESIGN RULE CHECK
NET LIST
PIN LIST
NET/PIN LIST CONVERSION
REPORT
LOTUS OUTPUT FILE
DBASE OUTPUT FILE
DELIMITED OUTPUT FILE
SDF OUTPUT FILE

SCHEMA-PCB PIN LIST
WIRE LIST
ROUTING CHECK
BUILD ALL
QUIT

Select the desired post-processor using the mouse or keyboard cursor keys, then press **<Enter>** or the mouse button. After the applicable post-processor is entered (or, if the post-processor(s) was entered when SCHPOST was invoked) SCHPOST begins its post-processing functions by reading the associated SCHEMA.CFG file for applicable configuration data. If SCHPOST is unable to locate or read the SCHEMA.CFG file, an error message is displayed and SCHPOST returns to the Manager or DOS.

Next, SCHPOST reads the schematic's associated DRAWING.CFG file, if applicable. Once SCHEMA.CFG and DRAWING.CFG are read, SCHPOST opens the drawing's object libraries assigned to the configuration file **LIBR1 - LIBR9** variables.

If the libraries can not be opened, an error message is displayed and SCHPOST returns to the Manager or DOS. Once the object libraries are opened, SCHPOST reads the schematic file(s) beginning with the first page (the file that has a **.P01** extension) and displays the message:

DRAWING.P01

where **DRAWING** is the drawing name. If the first schematic page file can not be opened (file does not exist or is not a SCHDRAW-produced schematic file) an error message is displayed and SCHPOST returns to the Manager or DOS. SCHPOST processes multi-page schematics page-by-page. Thus, a delay occurs between each page. As SCHPOST reads schematic pages the page number is incremented until all pages have been located and read. The message:

DRAWING.PXX

is displayed for each page processed. After all pages have been located and read, SCHPOST executes then merges and sorts the pages together to create the .PST (part list file) and the .NST (netlist file). These two files will be used to extract the selected post-processor outputs. As each post-processor executes, its output is directed to the preselected I/O device. If there are any routing errors then SCHPOST will prompt you to whether you wish to continue. It is highly recommended that all routing errors be corrected before using any generated output. Upon completion of all selected post-processing, the following prompt appears:

Press any key to continue ...

After a key is pressed, the SCHPOST options appear. Select QUIT and press **<Enter>** to return to the Manager. However, if SCHPOST is invoked from DOS, SCHPOST will automatically return to DOS.

The default mode of operartion for SCHPOST is the Incremental Compile Mode. The program will read the time/date stamp of the schematic pages and compare it to a corresponding, Oxx file, if it exists. If the Object file is newer than the schematic page then the page will not be re-processed.

WARNING: If you are transferring schematic pages from different PCs, be aware that the time/date stamp that one PC will create may not be synchronized to another system. For example, if one machine is a day or hour off from another then the incremental compile mode may not know to re-process a page. If there is any doubt use the **BUILD ALL** feature or load each page and save it to itself again to achieve a matching time/date stamp that is concurrent with your PC.

7.4 OUTPUT CONFIGURATIONS

Each SCHPOST post-processor generates an output listing. Five of the thirteen post-processor outputs are directed to the standard output device and may be redirected to any physical device. The remaining post-processors, the Net and Pin List post-processors, send their output to unique files.

Net list files have the same name as the input schematic file(s), with an extension of .NET, pin list - .NPL, Lotus - .WKS, Dbase - .DBF, Delimited - .TXT and SDF - .SDF. The net and pin list file formats are discussed in Appendix B.

The SCHEMA-PCB pin list is output to the \PCB\FILES subdirectory as two files: DRAWING.SCH and DRAWING.FWR.

7.4.1 Routing

The Routing post-processor output consists of a series of error messages. There are no headings. If no errors are located by the Routing post-processor, there is no Routing output.

Some Routing error messages are in the following form:

MESSAGE AT PAGE: DRAWING.PXX LOC: XXX.XX,YYY.YY

DRAWING.PXX is the particular schematic page where the error was found, LOC: XXX.XX,YYY.YY is the X- and Y-coordinate (Cursor Coordinate Window value) of the error, and MESSAGE is one of the following error messages:

DOT DOES NOT HAVE THREE CONNECTIONS

At least three intersecting signal wires/pin headers cannot be found at the applicable connection dot. If, upon examination of the error location, the junction appears correct, erase the dot and replace it with another dot. (An error occurred because the dot was not added last.).

FLOATING WIRE/LABEL

A signal wire or label has been found at the error location, i.e., the signal wire end point or label is floating and there is nothing connected to it. All signal wire end points and labels must terminate at another signal wire, label, or component pin header. Note that arcs do not constitute a legitimate connection. Examples of routing errors are contained in figure 7-1.

REFERENCE DESIGNATOR USED TWO OR MORE TIMES

This check outputs errors consisting of component pairs that have identical reference designators. If more than two components have the same reference designator, the components are displayed as multiple pairs. The error format is:

TEXT1,,,,,/NN,TEXT2,,,,,/MM

where: **TEXT1** and **TEXT2** are the SCHDRAW Schematic Editor **DRAW-OBJECT** Parameter Strings for the two applicable components and NN and MM are the schematic page numbers that contain the components. The **DRAW-OBJECT** Parameter Strings contain the component name, reference designator, and any other component parameters.

NETS WITH MORE THAN ONE LABEL

This error occurs when the user has placed two different labels for the same physically connected net. The first label is preceded by an '%' sign. Then all of the pins in the net are displayed. The format of each pin is as follows:

RRR-PD/PN

where: **RRR** is the object reference designator; **PD** is the pin definition containing both the pin type and pin name (**BODY-DF PI** entry); and PN is the page number where the object was located.

The second label appears at the end of the list, also with a preceding '%' sign.

7.4.2 Bill of Materials

The Bill of Materials (BOM) post-processor output is user-defined and can appear in almost any form, depending on the values assigned to variables **BOM1**, **BOM2**, and **BOM3** in the associated DRAWING.CFG or SCHEMA.CFG file. Please refer to Chapter 5, *SCHCFG*, for information on assigning the BOM variables.

The **BOM1** variable specifies the contents of the BOM header (title) line. The header line resides at the top of each BOM page and is preceded and succeeded by lines of dashes. The header line can be omitted from the BOM by not including the H() entry in the configuration file **BOM3** variable. A sample **BOM1** entry is:

BOM1=BILL OF MATERIALS PROCESSOR 100-0134-002 REV 3A PAGE

The BOM post-processor always positions the page number two spaces after the last character of the **BOM1** field. The page number begins with 1 on the first BOM page and is automatically incremented on

succeeding pages. If a <T> is included in the BOM1 field, the current time and date is placed in the header line (20 <Spaces> must follow the <T> entry).

```
FLOATING     WIRE     AT     PAGE:FIG7-1.P01
LOC:025.00,017.00
FLOATING     WIRE     AT     PAGE:FIG7-1.P01
LOC:027.00,017.00
DOT   DOES   NOT   HAVE   3   CONNECTIONS   AT
PAGE:FIG7-1.P01 LOC:025.04,017.01
FLOATING     WIRE     AT     PAGE:FIG7-1.P01
LOC:089.00,044.08
FLOATING  LABEL  (-SELECT2)  AT  PAGE:FIG7-
1.P01 LOC:089.04,044.08

NETS WITH MORE THAN ONE LABEL:
%-CLEAR        U1-I1/01   U2-I1/01   U3-I1/01
U4-I1/01      %PRESET

FLOATING     WIRE     AT     PAGE:FIG7-1.P02
LOC:056.00,016.08
FLOATING  LABEL  (-ENABLE2)  AT  PAGE:FIG7-
1.P02 LOC:056.02,016.07
FLOATING     WIRE     AT     PAGE:FIG7-1.P02
LOC:085.00,023.00
FLOATING LABEL (BUSB07) AT PAGE:FIG7-1.P02
LOC:084.10,023.00
ERROR:  REFERENCE  DESIGNATOR  USED  TWO  OR
MORE TIMES
PART:  74LS157\1A,U100,  PAGE:  1  -  THROWN
AWAY
FATAL ERRORS: PROCESS ABORT
```

Figure 7-1. Routing Error Examples

The **BOM2** variable specifies the contents of the column identifier (title) line. This line is positioned below the header line (on each page) and identifies the contents of each column. The column identifier line can be omitted from the BOM by not including the **H()** entry in the configuration file **BOM3** variable. A sample **BOM2** entry is:

BOM2=ITEM QUAN PART NUMBER DESCRIPTION REF DES

The **BOM3** field specifies part line contents and contains the combining and header line options. The **BOM3** variable consists of a series of entries separated by commas. The first entries specify fields, while the last two entries specify options. Each entry consists of an identifier, opening parenthesis, optional dash (-), number, and closing parenthesis. For example: **Q(10)**. The identifier denotes field type and consists of one of the following:

(Numbers in brackets are the maximum allowable field length.).

Dn (Description (DESC) field; where **n=1** to **4**) [64]

N (Item number; first line=**1**. It is automatically incremented for each succeeding line.) [8]

Q (Quantity; the number of parts this line represents) [8]

R (Reference Designator; the part's reference designator. Depending on the combine option, this field may contain one or multiple reference designators. Must be left-justified.) [64]

S (Space; adds blank spaces. This is the only **BOM3** field that can be used more than once in the **BOM3** line.) [16]

T (Text; Part Type, for this line.) [64]

The optional dash (-) specifies that field text should be left-justified, rather than default right-justified. The Reference Designator (R) field *must* be left-justified.

The numeric field specifies the field character length.

Option (combine and header) entries follow the field entries. Option entry formats are similar to the field entry format. Combine and header entries are defined as follows:

C(aa-bb) *Combine Fields.* If the combine field entry is present, the BOM output lines are sorted and combined prior to being output. If the combine field entry is not present, no sorting or combining occurs and raw output lines are output. Sorting and combining is done using character positions defined in the combine entry. **aa** is the starting character position. **bb** is the ending character position. Initially, all output lines are sorted by this **aa-bb** character field. Then, all output lines that have identical **aa-bb** character fields are combined into a single output line. The starting character position must begin with the first character of a field and the ending character position must end with the last character of a field. When combining output lines, reference designator fields from all combined lines are extracted and placed in the remaining line's reference designator field. If more reference designators are present than can fit into this single line's reference designator field, more output lines are created. These lines are empty except for the reference designator field.

H() *Header.* If present, header lines and identifiers (**BOM1** and **BOM2**) are included in the BOM output listing. If absent, **BOM1** and **BOM2** are not included in the BOM output listing and no page breaks are specified.

L(aa) *Length of Page*. This field is to identify the number of lines on the page used to print the Bill of Materials. The default is 66 lines. This field allows you to change the number of lines based on the size of paper being used.

Option entries are separated from field entries with two commas. For example:

BOM3=N(8),Q(8),S(2),D1(14),S(2),D2(-60),R(-16),,C(19-94),,H()

This BOM output line is 110 characters in length when all of the field entries are totaled. Header lines, identifiers, and page breaks will be output, **H()**. The output will be sorted and combined via characters in fields 19 through 94, **C(19-94)** (fields **D1** and **D2**). The actual BOM output line will consist of an item number in columns 1 through 8, **N(8)**; the quantity in columns 9 through 16, **Q(8)**; two blank spaces, **S(2)**; the Name Object Description 1 field (**DESC 1**) in columns 19 through 32, **D1(14)**; two more blank spaces, **S(2)**; a left-justified Name Object Description 2 field (**DESC 2**) in columns 35 through 94, **D2(-60)**; and left-justified reference designators in columns 95 through 110, **R(-16)**. Another example is: **BOM3=T(54),D1(14),D2(54)**

This example defines a BOM output line 122 characters in length. The BOM listing will not contain header lines and no combining will occur. The BOM output line will contain three fields: the component parameter string in columns 1 through 54, **T(54)**; the Name Object Description 1 (**DESC 1**) field in columns 55 through 68, **D1(14)**; and the Name Object Description 2 field (**DESC 2**) in columns 69 through 122, **D2(54)**. This is a typical example of a machine-readable BOM output. Figure 7-2 contains a sample BOM output listing produced from the following **BOM3** entry:

BOM3=N(8),Q(8),S(2),D1(14),S(2),D2(-60),R(-16),,C(19-94),,H()

BILL OF MATERIALS XYZ COMPUTER CORP. 4LATCH
150-0001-001 REV 1A 1

ITEM	QUAN.	PART NUMBER	DESCRIPTION	REF. DES.
1	1	001-0001-001	CONNECTOR	J1
2	4	050-0010-001	74LSXXX - QUAD NOTHING GATE	H1000/U3 H1001/U3 H1002/U3 H1003/U3
3	4	050-1000-025	RESISTOR,1K,	H1000/R1 H1001/R1 H1002/R1 H1003/R1
4	4	300-0000-001	74LS00 - LS TTL QUAD 2 INPUT NAND	H1000/U1 H1001/U1 H1002/U1 H1003/U1
5	4	300-0074-001	74LS74 - LS TTL DUAL D FLIP-FLOP	H1000/U2 H1001/U2 H1002/U2 H1003/U2

Figure 7-2. Sample BOM Listing

7.4.3 Design Rule Check

The Design Rule Check (DRC) post-processor output listing consists of per-page title blocks followed by six discrete checks. A sample DRC output listing is contained in figure 7-3.

The DRC title block format is also used by the Net List, Usage Reports, and Wire List post-processors. The first line of the title block contains the name of the applicable post-processor, followed by the current time, date, and page number. The second line contains, in order, the values assigned to the **F1, F2, F3,** and **F4** title block fields in the associated configuration file (DRAWING.CFG or SCHEMA.CFG).

The five discrete design rule checks are identified as:

FLOATING INPUTS
NETS WITH NO DRIVING SOURCE
NETS WITH MULTIPLE DRIVING SOURCES
NETS WITH A SINGLE PIN
PIN WITH MORE THAN ONE LABEL

Below each DRC check is a list of applicable errors. If no errors are located, a **NONE FOUND** message is listed below the applicable DRC check. If a given check is continued on the next page, a continuation message is displayed at the top of the applicable page, below the title block. For example:

FLOATING INPUTS (CONTINUED)

These errors are discussed in paragraph 7.2.3. The output format is discussed below.

The **FLOATING INPUTS** error message format consists of a list of objects/pin numbers configured as: **RRR-PD/PN**

```
-------------------------------------------------------------------
DESIGN RULE CHECK      Tue Mar 19 16:06:09 1991      PAGE 1
4LATCH SCHEMATIC  150-0001-001 REV 1A  7/06/88
-------------------------------------------------------------------


FLOATING INPUTS:
  NONE FOUND

NETS WITH NO DRIVING SOURCE:
%STROBE        H1000/U3-I1/01   H1001/U3-I1/01   H1002/U3-I1/01
                 H1003/U3-I1/01

NETS WITH MULTIPLE DRIVING SOURCES:
  NONE FOUND

NETS WITH A SINGLE PIN:
%DATA1         H1000/U1-I9/01
%DATA2         H1001/U1-I9/01
%DATA3         H1002/U1-I9/01
%DATA4         H1003/U1-I9/01

PIN WITH MULTIPLE NETS:
  NONE FOUND
```

Figure 7-3. Sample Design Rule Check Output Listing

where: **RRR** is the object reference designator; **PD** is the pin definition containing both the pin type and pin name (**BODY-DF PI** entry); and **PN** is the page number where the object was located.

The three **NETS WITH...** error message formats consist of a list of all nets that contain errors. Each net is identified by a net name (or label) and list of net components. The first net line contains the label and up to four components. If more than four components are present, they are listed on succeeding lines, four per line. For example:

%LABEL COMP1 COMP2 COMP3 COMP4
COMP5 COMP6 COMP7 etc.

LABEL is the net identifying label. If a net has multiple labels (an error condition), then **LABEL** is the first label found. **LABEL** is always preceded by a percent sign (%). If a net does not have a label, a net name is created by generating an eight digit number. The first two digits are the page number, the next three digits are the X coordinate and the last three digits are the Y coordinate. To find the approximate location of the net, pan the drawing until the X and Y coordinates are shown in the Cursor Coordinate Window. In figure 7-8 there is an autoassigned hierarchical net name which consists of a hierarchical prefix and an autoassigned net name, **H1000/01024012.** The H1000/ is the hierarchical prefix, the 01 is the page number, the 024 is the X coordinate and the 012 is the Y coordinate.

Each component, **COMPn**, is listed in the format:

RRR-PD/PN

where fields are identical to those described above under **FLOATING INPUTS.** If a net has multiple labels, remaining labels are listed as

components. Thus, any component beginning with a percent sign (%) is a label.

The **PINS WITH MORE THAN ONE LABEL** has the format:

REFD LABEL PIN DEFINTION/PAGE NUMBER

7.4.4 Net and Pin Lists

The Net and Pin List post-processor output file formats are discussed in Appendix B. The SCHEMA-PCB pin list is output to the \PCB\FILES subdirectory as two files: DRAWING.SCH AND DRAWING.FWR.

7.4.5 Wire List

The Wire List post-processor output identifies all nets used in the schematic. The list is alphabetized by label. Figure 7-4 contains a sample wire list output.

WIRE LIST Tue Mar 19 16:06:35 1991 PAGE 1
4LATCH SCHEMATIC 150-0001-001 REV 1A 7/06/88

---

1 +5 H1000/R1-L H1000/U1-14 H1000/U2-14
H1000/U3-16 H1001/R1-L H1001/U1-14
H1001/U2-14 H1001/U3-16 H1002/R1-L
H1002/U1-14 H1002/U2-14 H1002/U3-16
H1003/R1-L H1003/U1-14 H1003/U2-14
H1003/U3-16

2 DATA1 H1000/U1-9

3 DATA2 H1001/U1-9

4 DATA3 H1002/U1-9

5 DATA4 H1003/U1-9

6 GND H1000/U1-10 H1000/U1-7 H1000/U2-7
H1000/U3-8 H1001/U1-10 H1001/U1-7
H1001/U2-7 H1001/U3-8 H1002/U1-10
H1002/U1-7 H1002/U2-7 H1002/U3-8
H1003/U1-10 H1003/U1-7 H1003/U2-7
H1003/U3-8

7 H1000/01036020 H1000/R1-R H1000/U2-1 H1000/U2-4

8 H1000/01036026 H1000/U1-8 H1000/U2-2

9 H1000/01036032 H1000/U2-3 H1000/U3-2

10 H1001/01036020 H1001/R1-R H1001/U2-1 H1001/U2-4

```
11   H1001/01036026  H1001/U1-8    H1001/U2-2

12   H1001/01036032  H1001/U2-3    H1001/U3-2

13   H1002/01036020  H1002/R1-R    H1002/U2-1    H1002/U2-4

14   H1002/01036026  H1002/U1-8    H1002/U2-2

15   H1002/01036032  H1002/U2-3    H1002/U3-2

16   H1003/01036020  H1003/R1-R    H1003/U2-1    H1003/U2-4

17   H1003/01036026  H1003/U1-8    H1003/U2-2

18   H1003/01036032  H1003/U2-3    H1003/U3-2

19   OUT1        H1000/U2-5    J1-1

-------------------------------------------------------------------------
 WIRE LIST              Tue Mar 19 16:06:36 1991      PAGE 2
 4LATCH SCHEMATIC  150-0001-001 REV 1A  7/06/88
-------------------------------------------------------------------------
20   OUT2        H1001/U2-5    J1-2

21   OUT3        H1002/U2-5    J1-3

22   OUT4        H1003/U2-5    J1-4

23   STROBE      H1000/U3-1    H1001/U3-1    H1002/U3-1
                 H1003/U3-1
```

Figure 7-4. Sample Wire List Output

The output listing contains the standard post-processor title block (see paragraph 7.4.3) followed by a list of individual nets. A blank line separates each net.

Net format is:

NUM LABEL COMP1 COMP2 COMP3 COMP4 COMP5 COMP6 etc.

where NUM is an assigned net number. The first net is assigned number 1 and succeeding nets are automatically incremented. LABEL is the net identifying label. If a net does not have a label, a net name is automatically created by generating an eight digit number as described in the section on Design Rule Check.

Each component, COMPn, consists of a reference designator, dash (-), and pin name.

If a net has multiple labels (an error identified by the Design Rule Check post-processor), the first label is used as LABEL and remaining labels are not listed.

If a net contains only one component, it is not considered a net and thus, is not listed.

7.4.6 Reports

The Reports post-processor output consists of signal label and reference designator usage reports. Each report is generated separately. Figures 7-5 and 7-6 contain sample signal label usage and reference designator usage reports, respectively.

The signal label usage report lists the standard post-processor title block (see paragraph 7.4.3) followed by an alphabetical listing of all signal labels. A listing of the schematic pages that the signal labels appear on is also displayed with the labels.

The reference designator usage report also uses the standard post-processor title block and lists all reference designators in alphabetical order. On the same line as each reference designator is the applicable object name and a list of sub-types used (A, B, C, etc.). Included with each sub-type is the applicable page number. Objects without sub-types are identified as sub-type A.

```
------------------------------------------------------------------------
 REPORT LABEL          Tue Mar 19 16:04:25 1991      PAGE 1
 4LATCH SCHEMATIC  150-0001-001 REV 1A  7/06/88
------------------------------------------------------------------------


+5          01
DATA1       01
DATA2       01
DATA3       01
DATA4       01
GND         01
OUT1        01
OUT2        01
OUT3        01
OUT4        01
STROBE      01
```

Figure 7-5. Sample Label Usage Report

```
---------------------------------------------------------------------
REPORT REF. DES.        Tue Mar 19 16:04:24 1991      PAGE 1
4LATCH SCHEMATIC  150-0001-001 REV 1A  7/06/88
---------------------------------------------------------------------

H1000/R1        DRES        A/01
H1000/U1        74LS00      C/01
H1000/U2        74LS74      A/01
H1000/U3        74LSXXX     A/01
H1001/R1        DRES        A/01
H1001/U1        74LS00      C/01
H1001/U2        74LS74      A/01
H1001/U3        74LSXXX     A/01
H1002/R1        DRES        A/01
H1002/U1        74LS00      C/01
H1002/U2        74LS74      A/01
H1002/U3        74LSXXX     A/01
H1003/R1        DRES        A/01
H1003/U1        74LS00      C/01
H1003/U2        74LS74      A/01
H1003/U3        74LSXXX     A/01
J1              ^ ^CONN     A/01,A/01,A/01,A/01
```

Figure 7-6. Sample Reference Designator Usage Report

7.4.7 Label Swapping

In some instsances the user may wish to swap or exchange one label for another. For example, some of the analog parts that are supplied with SCHEMA have the hidden power pin assigned to the net, VCC. This is because the supply voltage has a wider range of possible values. If we then wish to have VCC become the net name +12 we will use the LBLSWAP object. This is a special object similiar to the GLOBAL object used for hierarchical design dicussed later. This object must be placed on the first page of the schematic. If there are LBLSWAP objects in the schematic, SCHPOST will create a file, DRAWING.LBL. The following is how to enter the part on the DRAW-OBJECT command line:

LBLSWAP,,VCC,+12 <Enter>

This will cause the label VCC to disappear and be replaced or merged with the net, +12.

Note: Be sure to give no reference designator to the object by typing two commas consecutively.

7.5 HIERARCHY

Hierarchy is a feature that is useful in creating schematics that are designed from top down. The design process is simplified by representing familiar circuits as modules or blocks in the schematic. If there is a particular circuit that is used repeatedly throughout a schematic, that circuit can be replaced by a single block at every point that it occurs. These blocks are called **Hierarchical Objects** and the circuits that these blocks represent are called **Hierarchical Pages**. For example, if a particular voltage driver circuit at several points in a schematic is needed, instead of placing the entire driver circuit at all the different points, just insert a single hierarchical object representing that circuit.

Hierarchical objects are linked to their associated hierarchical pages through part type name and pin definitions. The part type name of a hierarchical object corresponds to the page name of a schematic. This schematic contains the circuit that is represented by the hierarchical object in the main, or top level, schematic. Figure 7-7 is an example of a hierarchical object that has the part type name, **LATCH** and the first page of the associated schematic, **LATCH.P01**.

The pin definitions of a hierarchical object are the link between the nets in the top level schematic and the nets in the hierarchical circuit. The name of the pin definition corresponds directly to a net with that same name in the hierarchical page. As illustrated in figure 7-7, a pin is defined on the hierarchical object as **CLK** and a net on the hierarchical page is also labeled **CLK**. The net connected to the pin **CLK** on the top-level, in this case **STROBE**, actually contains the net **CLK** that is at the hierarchical level. The name of the completed net is the name of the net at the top level. Figure 7-7 shows the net traversing through pin **CLK**, into net **CLK** on the hierarchical page is called **STROBE**.

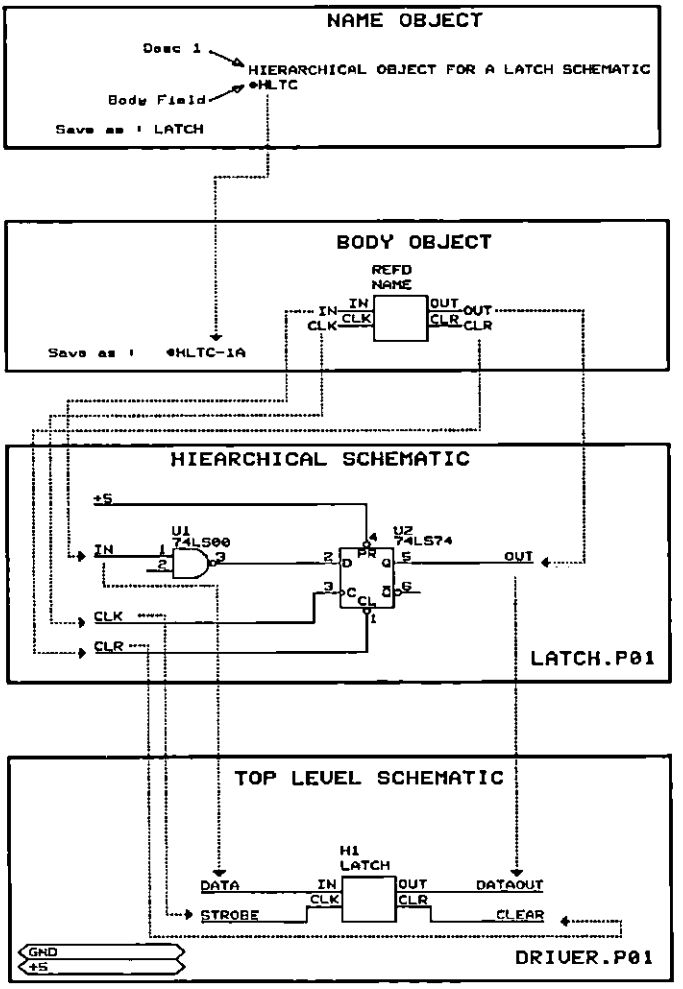


Figure 7-7. Hierarchical Illustration

The following sample netlist is from the previous example. Notice that when the schematic is flattened the intermediate net names of the hierarcical object, **IN, CLK, CLR,** and **OUT** dissappear. **+5** remains, because it was defined as global. The reference designators of the two parts are preceded with a **H1000/** . If there were more than one hierarchical object than the two different prefices would identify from which hierarchical object they originate. The original reference designator given to the hierarchical object, **H1,** disappears. There is a hierarchical map file with the extension **.HSF,** that shows the key to which prefix was assigned to which hierarchical object. For this example, the **DRIVER.HSF** file will have the one entry:

```
1000    H1\LATCH
```

```
[
74LS00
H1000/U1
300-0000-001
74LS00 - LS TTL QUAD TWO INPUT NAND

]
[
74LS74
H1000/U2
300-0074-001
74LS74 - LS TTL DUAL D FLIP-FLOP

]
(
+5
H1000/U1-14
H1000/U2-14
```

```
H1000/U2-4
)
(
CLEAR
H1000/U2-1
)
(
DATA
H1000/U1-1
)
(
DATAOUT
H1000/U2-5
)
(
GND
H1000/U1-7
H1000/U2-7
)
(
H1000/01016012
H1000/U1-3
H1000/U2-2
)
(
STROBE
H1000/U2-3
)
```

Figure 7-8. Sample Hierarchical Net List

When SCHPOST is run on a schematic containing hierarchical objects, the connections to those objects are followed through to the corresponding hierarcical pages. As far as SCHPOST is concerned, the hierarchical objects are replaced by their corresponding circuits and the hierarchical objects themselves are invisible. In other words, hierarchical objects are transparent to SCHPOST. See the sample net list on figure 7-8 for clarification.

Notice in the sample net list that there have been prefices (Hxxx/) added to the reference designators of all the parts in the hierarchical page. The H denotes the part as being in a hierarchical page, while the number that follows tells on which hierarchical page the part is contained. The hierarchical prefices are numbered in a linear fashion, as each hierarchical object is encountered by SCHPOST. Since each part in a schematic must have a unique reference designator, each hierarchical page must have its own prefix number, even if the page is repeated. The hierarchical prefix is also added to the name of each net that is internal to the hierarchical schematic.

A feature of hierarchy, that is not shown in the example, is the ability to nest hierarchical objects. It is possible to have a hierarchical object within the circuit of another hierarchical object. There is no set level of nesting, but the schematic representing the hierarchy is limited to 99 pages. Below is a set of rules for creating hierarchical objects.

7.5.1 Hierarchical Design Guidelines

1. The Body Object name for all hierarchical objects must begin with #H. See the paragraph 1.5.4.6.2.1 in Chapter 1, *Getting Started* for instructions on making a hierarchical object.

2. The Name Object name for all hierarchical objects must have a corresponding hierarchical schematic of the same name that shows the logic of the hierarchical object.

3. The pin definitions on hierarchical objects should match the net labels on the hierarchical schematic that corresponds to the object. When the top-level schematic is flattened, the signals going into the pins of the hierarchical object are actually going into the nets on the corresponding hierarchical schematic.

4. The hierarchical schematic must be in the same sub-directory as the library that contains the hierarchical object. It can also be in the current directory.

5. When designing schematics for the purpose of PCB layout, hierarchical objects/drawings cannot be used repetitively. In other words, a particular hierarchical object/drawing can only be used one time within a schematic. The reason for this constraint is that for any forward or back annotation to be possible, every unique part on the PCB must have a corresponding unique part in the schematic. If a particular hierarchical object/drawing needs to be used multiple times, the drawing and object can be saved to a different name for each time that they are used. The 'Z' option will eliminate the hierachical prefix from the beginning of reference designators.

7.5.2 Global Nets

Nets exist at different hierarchical levels, without being passed down through a hierarchy, are called **Global Nets.** Global nets must be designated as such at the top level of the hierarchy. Whenever an interior net (a net that does not connect outside that level) is found in a hierarchical page, that net is checked against the global nets declared at the top level. If that net name matches one of the declared global nets, then a hierarchical prefix will not be added to that net name and anything connecting to that net will connect to everything in the associated global net. Refer to figure 7-7, Hierarchical Illustration, for an example of global nets.

7.5.2.1 Global Net Guidelines

1. Global Nets must be declared using the global net object called GLOBAL. The global net object is loaded like a normal object, except that there is no reference designator and parameter 1 contains the name of the net being made global. For example, from the DRAW-OBJECT text window type: **GLOBAL,,NETNAME <Enter>.**

2. All Global Nets must be declared on the first page of a top level schematic. A global net declaration on any other page will result in an error message.

3. All global nets must be specifically declared, including **+5** and **GND.** If either **+5** or **GND** is used in a hierarchical page, but not declared as a global variable, a warning will be generated.

Refer to section 1.5.4.6.2.3 in Chapter 1, *Getting Started* for instructions on declaring a global net.

7.5.3 Multiple Configuration Files

Since hierarchy involves multiple schematics, the user can have multiple configuration files. Typically the user will want to use only one configuration file. This would be the Top level schematic configuration file. However, if you wish to have a special configuation file for a lower level schematic you may do so. If a lower level config file is not found then the Top level DRAWING.CFG or the main SCHEMA.CFG will be used.

7.6 CONVERSION PROGRAMS

The Net/Pin List Conversion (C) option in SCHPOST invokes the conversion programs, CONVERT.EXE. The following formats are available:

CBDS Format IBM Mainframe
CADNETIX / SCI-CARDs
CALAY
Computer Vision
dataCON Wire-Wrapping
EDIF 1 0 0 Netlist
EDIF 2 0 0 Netlist
EDIF 2 0 0 Flat for PlusLogic
Visionics EE Designer
FutureNet Netlist
FutureNet Pinlist
Intergraph
PCAD-PDIF
Racal Redac
Redac Interface Format (RINF)
Redac CADSTAR Pro Body/Component List
PSpice
Telesis

See Appendix B, *Net/Pin List Formats* and Chapter 8, *Utilities.* There are some BASIC versions of the formats supplied for you on the distribution disks in the NETLIST.ZIP file. Use the PKUNZIP.EXE to recover them.

7.7 VIRTUAL MEMORY

Virtual Memory is a process that uses the computer's hard disk to store information that would normally be stored in memory. SCHPOST uses this process by storing schematic page data to the hard disk while each page of the schematic is being processed. When all of the pages have been read, the information is retrieved from the hard disk to produce the requested SCHPOST option, such as Bill of Materials.

Since most of the information is saved to disk instead of memory, SCHPOST can process schematics much larger than it could without this feature. In order to avoid delays caused by excessive disk activity, information is not written to disk until all of the available memory has been used. All the Merging and Sorting of the the individual pinlist files, O01, O02, etc. are all processed on disk. This will increase the speed of processing a design.

If you do want the disk information saved to another directory or a Ram Drive you will need to use the SCHTEMP environment variable. For example:

SET SCHTEMP=D:\ (where D: could be a Ram Drive)

or

SET SCHTEMP=C:\TEMP (where C:\TEMP is where all the process files are sent)

All the intermediate merge and sort files along with the .NST and .PST files will be sent to this path.

Chapter 8. Utilities

8.1 INTRODUCTION

This chapter describes SCHEMA III's utility programs. The utility programs and their functions are:

a. **SCHBACK** - Provides third-party (non SCHEMA-PCB) PCB layout back annotation of component reference designators and gate sections.
b. **SCHECO** - Used to forward annotate SCHEMA III schematic design changes to the SCHEMA-PCB board layout and back annotate SCHEMA-PCB board design changes to the SCHEMA III schematic.
c. **SCHLOCAL** - Combines all objects used in a drawing set into one, local object library.
d. **SCHPAGE** - Provides virtual page creation.
e. **Net/Pin List Conversion Programs** - Convert SCHEMA III net/pin lists to other popular CAE system formats.
f. PSpice Conversion.
g. Network Capatibility

8.2 SCHBACK

SCHBACK is used to back annotate SCHEMA III reference designators/gate sections automatically from a text file. (*SCHECO is used to back annotate SCHEMA-PCB data.*)

As schematics are created with SCHEMA III, reference designators are assigned to components. After drawings are submitted to a third party PCB layout system (non SCHEMA-PCB), the reference designators are usually changed to reflect the physical component layout. During the layout process, component gate sections may also change. SCHBACK provides the capability to update all reference designators/gate sections in a schematic from a common 'was/is' data file.

The 'was/is' file is an ASCII file containing reference designator/gate section data for each reference designator/gate section in the schematic. Each line of the ASCII 'was/is' file contains a pair of reference designators separated by a comma and one or more spaces. The reference designator left of the comma is the current ('was') reference designator, while the reference designator right of the comma is the new ('is') changed reference designator. Optional gate section entries are separated from applicable reference designators with a backslash character. For example:

.AAA01, U1 (Component **.AAA01** is now **U1**)
U89\C, U10\D (Component **U89**, gate section **C** is now **U10**, gate section **D**).

If a reference designator and/or a gate section need to be swapped with another reference designator and/or adjacent gate section, then two adjacent entries must be entered in the 'was is' file. For example:

U1, U2 (every **U1** will be swapped for a **U2**)
U2, U1 (every **U2** will be swapped for a **U1**)

The ASCII 'was/is' file can be created using any text editor, word processor, or perhaps automatically from the associated PCB layout system. When SCHBACK runs, it processes data from the ASCII 'was/is' file and updates each schematic page.

The ASCII 'was/is' file carries the same prefix filename as the applicable schematic file, with a suffix file extension of .REF. A sample ASCII 'was/is' file, TUTORIAL.REF, is included with SCHEMA III. The following paragraphs explain how to invoke SCHBACK and use the TUTORIAL.REF file in a tutorial exercise.

The hierarchical SCHBACK file is an ASCII file containing reference designator/gate section data for each hierarchical reference designator/gate section in the schematic. Each line of the hierarchical SCHBACK file (.HBK) contains a reference designator followed by one or more spaces and a hierarchical schematic file name with a complete path. The reference designator on the left is the reference designator of an object located on a hierarchical drawing, while the file name on the right is the schematic drawing name of where the object is located. For example:

U3 \DRWNG\LOGIC.P01 (Hierarchical Component U3 is located on the schematic page LOGIC.P01 in subdirectory \DRWNG)

The hierarchical SCHBACK file (.HBK) can be created using any text editor, word processor or automatically created when SCHBACK invokes the post processor, SCHPOST. When SCHBACK runs, it processes the hierarchical SCHBACK file (.HBK) to determine which of the reference designators represent hierarchical objects as well as what schematic page these hierarchical objects reside in.

The hierarchical SCHBACK file is an ASCII file that carries the same prefix filename as the applicable schematic file, with a suffix file extension of .HBK.

8.2.1 Invoking SCHBACK

SCHBACK can be invoked from the Manager or directly from the DOS prompt. Before SCHBACK is invoked, DOS must be booted. Hard disk users should make the SCHEMA directory the current directory by entering:

CD \SCHEMA <Enter>

from the DOS prompt. Floppy disk users should place the Program disk in Drive A, the Drawing disk in Drive B, and make Drive A the current drive.

The SCHBACK command line is:

SCHBACK DRAWING OPTION

where **DRAWING** is the name of the selected schematic drawing to back annotate and **OPTION** denotes the SCHBACK options. If a **C** entry is included in the **OPTION** field, a check run of SCHBACK is initiated. With a check run, the schematic is processed normally, but the original schematic drawing pages are not changed. The C option is usually invoked to detect errors in the associated DRAWING.REF ASCII 'was/is' file. If a N is included in the **OPTION** field, the post processor will not be run. Normally, the post processor will be invoked by the SCHBACK utility in order to produce the hierarchical SCHBACK file (.HBK). However, if hierarchcial objects are not present in the ASCII 'was/is' file the N option should be used to speed up the SCHBACK operation by bypassing the post processor step.

NOTE

If the utility SCHPOST issues a 'NOT ENOUGH MEMORY, ABORTED' error message, then run SCHPOST using the 'H' option to create the

hierarchical SCHBACK file (.HBK). Next, run SCHBACK using 'N' option since the .HBK file has already been created through SCHPOST.

When SCHBACK executes, (unless the N option was specified), it invokes the post processor SCHPOST in order to generate the hierarchical SCHBACK file (.HBK). The hierarchical SCHBACK file (.HBK) is read, if present. Then the DRAWING.REF ASCII 'was/is' file is read. Next, SCHBACK reads the the first page of the schematic to process (DRAWING.P01), looks at each reference designator/gate section in the file and changes the reference designators/gate sections per the DRAWING.REF ASCII 'was/is' file. If a matching reference designator can not be found in the DRAWING.REF ASCII 'was/is' file, an error is reported. After SCHBACK finishes with the DRAWING.P01 page, it continues by reading the DRAWING.P02 page, and so on. If hierarchical information is present, then SCHBACK will continue reading and processing all of the hierarchical objects and files present in the hierarchical SCHBACK file (.HBK).

When all schematic pages have been processed, SCHBACK checks the DRAWING.REF ASCII 'was/is' file. It then outputs a list of all reference designator/gate section pairs from the ASCII 'was/is' file that were not referenced on any schematic page.

When SCHBACK updates schematic drawing files, it renames the original drawing files to DRAWING.Bxx and names updated files as DRAWING.Pxx. This procedure is identical to SCHDRAW. As previously mentioned, the C option can be included in the SCHBACK command line to check for proper back annotation without actually changing the original schematic drawing files.

8.2.2 SCHBACK Tutorial

For the SCHBACK tutorial exercise, we will back annotate the schematic pages named TUTORIAL.P01 and TUTORIAL.P02 with the TUTORIAL.REF file.

Before proceeding, print the TUTORIAL.REF and TUTORIAL.P01 files.

To run the SCHBACK tutorial, ensure the procedure at the beginning of paragraph 8.2.1 has been followed. Now invoke SCHBACK with the TUTORIAL schematic.

After SCHBACK executes, use SCHDRAW to compare the new TUTORIAL.P01 drawing page against the printed TUTORIAL.P01 and TUTORIAL.REF files. Notice that all component reference designators and gate sections in the new TUTORIAL.P01 drawing were updated according to the TUTORIAL.REF file.

8.2.3 SCHBACK User Messages

The following user messages are associated with SCHBACK:

NOT FOUND IN TABLE: <refd> - This is an error message. The reference designator **<refd>** could not be located in the DRAWING.REF ASCII 'was/is' file table. The reference designator is probably missing or it has an incorrect gate section identifier. The **<refd>** reference designator was not updated.

PROCESSING PAGE: <filename.Pxx> - This is a user message (not an error). Page **<filename.Pxx>** is currently being processed by SCHBACK.

TABLE ENTRY NOT USED: <was, is> - The **<was, is>** pair from the DRAWING.REF ASCII 'was/is' file was not used in any replacement. This is probably an error unless you knowingly have extra reference designator pairs in your DRAWING.REF ASCII 'was/is' file.

UNABLE TO OPEN HIERARCHICAL FILE - The hierarchical page file cannot be opened. Make sure that the path specified in the hierarchical SCHBACK file (**.HBK**) is correct for the location of that hierarchical schematic. Also check the **FILES=** statement in **CONFIG.SYS**.

8.3 SCHECO

SCHECO is used to forward annotate SCHEMA-PCB board layout designs with SCHEMA III schematic changes and back annotate SCHEMA III schematics with SCHEMA-PCB board design changes.

Chapter 9, *ECO*, of the *SCHEMA-PCB Layout User's Manual* contains more information on forward and back annotation. Please read Chapter 9 before using SCHECO.

8.3.1 Invoking SCHECO

Like most SCHEMA III programs, SCHECO can be invoked from the Manager or directly from the DOS prompt. Chapter 2, *SCHEMA Program Manager*, contains detailed information on the Manager and the procedure to invoke SCHECO.

Whether you use the Manager or DOS, the SCHECO command line is:

SCHECO DRAWING <Enter>

where DRAWING is the name of the selected schematic drawing process. Press **<Enter>** to invoke SCHECO.

8.3.2 Running SCHECO

After SCHECO is invoked, the screen is cleared and the following message is displayed:

To run FORWARD annotation press F <Enter>
To run BACK annotation press B <Enter>
To return to the SCHEMA MANAGER press Q <Enter>

When forward annotation (**F <Enter>**) is selected, SCHEMA III searches the \PCB\FILES subdirectory for file DRAWING.$$$, which represents the current SCHEMA-PCB board database. If the DRAWING.$$$ file is not found, an error message is output and control is returned to the Manager. If the DRAWING.$$$ file is found, a pin list is created from the preselected SCHEMA III drawing.

After the pin list file is created, it is compared against the DRAWING.$$$ file. After the files are compared, two new files are created: DRAWING.FWR and DRAWING.ECO. The files are then copied into the \PCB\FILES subdirectory and control is returned to the Manager. Next, SCHEMA-PCB is invoked and the SCHEMA-PCB **ECO-from Sch** command is used to load the DRAWING.ECO file. When SCHEMA-PCB loads the DRAWING.ECO file, the DRAWING.FWR file is renamed as DRAWING.$$$, to reflect the current board database.

Before back annotation can be done, you must first generate the SCHEMA-PCB back annotation file, using the SCHEMA-PCB **ECO-To Sch** command. Refer to Chapter 9, *ECO*, of the *SCHEMA PCB Layout User's Manual*, for additional information.

NOTE

Only reference designator and gate section changes made to the SCHEMA-PCB design can be back annotated to the SCHEMA III schematic. Any other changes are disallowed by SCHECO. Back annotation must only be done when the SCHEMA III schematic and SCHEMA-PCB board design are in complete logical agreement.

When back annotation (**B <Enter>**) is selected, SCHEMA III searches the \PCB\FILES subdirectory for the SCHEMA-PCB **ECO-To Sch** back annotation file, DRAWING.BCK. When DRAWING.BCK is found, SCHECO creates a 'was/is' file (named DRAWING.REF) of the reference designator/gate section changes. The DRAWING.REF file is placed in the current directory. If the DRAWING.BCK contains changes other than reference designator/gate section, an ASCII exception file is created as file DRAWING.EXP in the current directory. The DRAWING.EXP file lists all changes that can not be done by SCHECO. Next, SCHBACK is automatically invoked with the DRAWING.REF file to update the current SCHEMA III drawing. After the drawing is updated, control is returned to the Manager.

8.4 SCHLOCAL

SCHLOCAL automatically places all objects used in a given schematic in the library assigned to the schematic's configuration file **LIBR1** variable. When invoked SCHLOCAL checks each object of the associated schematic to see if the object is already in the library assigned to the drawing's configuration file **LIBR1** variable. If so, SCHLOCAL continues. If the object is not in the **LIBR1** library, SCHLOCAL copies the object into the **LIBR1** library. This process continues for each object used in the entire drawing. Upon completion, all of the drawing's objects reside in the **LIBR1** library.

8.4.1 Invoking SCHLOCAL

SCHLOCAL can be invoked from the Manager or directly from the DOS prompt. Before SCHLOCAL is invoked, DOS must be booted.

Hard disk users should make the SCHEMA directory the current directory by entering:

CD \SCHEMA <Enter>

from the DOS prompt.

Floppy disk users should place the Program disk in Drive A, the Drawing disk in Drive B, and make Drive A the current drive. The SCHLOCAL command line is:

SCHLOCAL DRAWING

where **DRAWING** is the name of the selected schematic drawing to process. Press **<Enter>** to invoke SCHLOCAL. Before invoking SCHLOCAL, ensure the desired library (normally a newly-created, empty library) is assigned to the associated configuration file's **LIBR1** variable.

8.4.2 Running SCHLOCAL

After SCHLOCAL is invoked, the screen is cleared and a program copyright notice message is displayed along the top of the screen. All objects used in the selected drawing are placed in the library assigned to the drawing's configuration file **LIBR1** variable. Control is then returned to the Manager or DOS.

8.5 SCHPAGE

SCHPAGE lets you create non-standard size blank pages. Most users will never use this program but some users may want the flexibility to create non-standard pages. For example, European users can use SCHPAGE to create European-standard engineering pages for SCHEMA.

All SCHEMA III pages are measured internally as X-dots by Y-dots. The actual *physical* output print size of a given page depends on the printer itself, but generally, printing occurs at 120 dots per inch (dpi) in the Normal print mode. Therefore, a page that is 1600x1200 dots creates a print about 14x10 inches. Appendix E, *Hardware Requirements*, lists the drawing page dpi densities for each supported printer and print mode used.

The actual *physical* output plot size of a given page depends on the PXLEN and PYLEN configuration parameters. The page is scaled to fit within the area specified regardless of the drawing page size.

SCHPAGE displays the internal page size (in dots) of existing pages and lets you create new blank pages of any size. The upper page size limit is directly proportional to the amount of free memory in your PC. The standard BPAGE3.P99 drawing page is close to the page size limit for PCs with 640K of memory (>590K *free* memory). For PCs with *expanded memory* installed, a comparison will occur between the amount of *free* RAM and *expanded* memory. The greater of the two will be used for the drawing. All expanded memory boards used must follow the LIM standard (LOTUS-INTEL-MICROSOFT).

The amount of FREE memory required for SCHDRAW to be able to load the page is given as follows (all sizes are approximate):

	Standard Memory	Standard/Expanded
A (640 x 480)	208,400	170,000 / 38,400
B (992 x 640)	249,360	170,000 / 79,360
C (1312 x 992)	332,688	170,000 / 162,688
D (2016 x 1312)	500,624	170,000 / 330,624
E (2624 x 2016)	---------	170,000 / 661,248

8.5.1 Invoking SCHPAGE

SCHPAGE can be invoked from the Manager or directly from the DOS prompt.

Before SCHPAGE is invoked, DOS must be booted. Hard disk users should make the SCHEMA directory the current directory by entering:

CD \SCHEMA <Enter>
SCHPAGE <Enter>

from the DOS prompt. Floppy disk users should place the Program disk in Drive A and the Drawing disk in Drive B. Next, from the DOS prompt enter:

A:SCHPAGE <Enter>

There are no file names or other parameters required to invoke SCHPAGE.

8.5.2 Running SCHPAGE

After SCHPAGE is invoked, the following message is displayed along the top of the screen:

SCHPAGE Version X.XX Copyright Omation Inc 1989
Enter:
D - Display X and Y values for a page
C - Create a new blank page
Q - Return to DOS (or Manager)

(where: **X.XX** is the current version number).

To display the X and Y dot (pixel) values for a page, enter **D**. After **D** is entered, the following prompt is displayed:

Enter the name of the drawing to display:

Enter the proper path (if the drawing is not in the current directory) and the complete drawing name including page extension (**.Pxx**). After the drawing name is entered, the following is displayed:

X: VALUE
Y: VALUE

where **VALUE** is the applicable page size in dots (pixels).

To create a new blank page, enter **C**. After **C** is entered, the following prompt is displayed:

Enter X page size in dots (must be a multiple of 32):

Enter the desired X value. After the X value is entered, the following prompt is displayed:

Enter Y page size in dots (must be a multiple of 16):

Enter the desired Y value. After the Y value is entered, the following prompt is displayed:

Enter the name of the new blank drawing page:

Enter the name of the new blank drawing page in the format:

B(P)PAGEX.P99 <Enter>

where B specifies Border or P specifies a completely blank page and X is any number greater than 3. For example, **BPAGE4.P99.** After the drawing name is entered, the following message is displayed:

File created ok

To exit SCHPAGE and return to DOS, enter Q. Control is returned to DOS or the Manager.

Note : The user can use SCHPAGE to make a custom page for printing or plotting. See Chapter 4 - *SCHPLOT\SCHPRINT.*

8.6 NET/PIN LIST CONVERSION PROGRAMS

In *ADDITION* to CONVERT.EXE, which contains all of the conversion utilities through SCHPOST, Omation has provided several BASIC programs in case a different output format is desired. These BASIC programs are included with SCHEMA III to convert SCHEMA III's net/pin list output to be compatible with other popular CAE systems. They are archived in the file NETLIST.ZIP on the distribution disks. The programs are:

CADNETIX.BAS - Cadnetix/SCI-CARDs Net List
CALAY.BAS - Calay Net List
CBDS.BAS - CBDS Net List**
CV.BAS - Computervision Net List
DATACON.BAS - dataCon Wire-Wrapping Net List
EDIF.BAS - EDIF Net List
EE.BAS - Visionics (EE Designer) Net List
FNNET.BAS - FutureNet Net List**
FNPIN.BAS - FutureNet Pin List**
IG.BAS - Intergraph Net List**
PDIF.BAS - P-CAD PDIF Net List**
RR.BAS - Racal-Redac Net List
SPICE.BAS - SPICE Pin List**
TELESIS.BAS - Telesis Net List

Programs marked with ** use the SCHPOST pin list output as their input, while remaining programs use SCHPOST's net list output as input.

All conversion programs require GWBASIC/BASICA or equivalent to run. Basically, the conversion programs read the input file (SCHEMA III net or pin list) and create an output file in the appropriate format. Additional information covering usage and output filenames is contained in individual program comments.

All the BASIC conversion programs pass labels, reference designators and pin numbers from the input file to the output file *unchanged* (i.e., only the format changes). Thus, care must be exercised to ensure any unusable characters, naming conventions, etc. that the end system requires have been adhered to in creating the schematic and any components.

The SCHEMA III net and pin list output formats are discussed in detail in Appendix B.

8.6.1 Net List Suggestions

When passing component outline types to a PCB layout system or passing simulation type fields to a logic simulation system, the SCHDRAW Object Editor NAME-DESC 3 and NAME-DESC 4 fields can be used to carry the applicable information.

8.6.2 Using SCHEMA III With PSpice (Analog)

SCHEMA III's SPICE pin list conversion program, SPICE.BAS, converts SCHEMA III's pin list output to a SPICE-compatible net list. However, the PSpice conversion found within the CONVERT.EXE program is a superset of the SPICE format that has been enhanced to output to PSpice in more detail than the SPICE.BAS. PSpice for Digital/Analog is discussed in the next section. The SPICE net lists in general are rather unique as far as reference designators, label usage and other data, so the information presented in the following paragraphs should be read carefully and adhered to when creating schematics for PSpice. Earlier versions of PSpice and Berkley SPICE requires that all node names are to be numeric and range from 0 to 9999. Therefore, all user-entered labels must be numeric. Also, the SCHPOST-created, automatically assigned labels for unnamed nets cannot be used. For example, 01036032 cannot be used, because it exceeds the maximum range of accepted integers. Instead, the net numbers, as listed in SCHPOST's wire list output, are used with 1000 added to them. The criteria for determining whether the label is auto-assigned is whether or not all of the characters are numeric and there are eight digits. For example, the auto-assigned net number 12 called 01042030 would be

node number 1012 in the PSpice net list file. To avoid, duplication, user-supplied labels (node names) should be less than 1000.

PSpice also uses the first letter of the component's reference designator to determine the part type. Therefore, never use auto-assigned reference designators as they will be incorrect. There are several library objects to be used with PSpice and are found in the ANALOG.LBR library. They are described in Table 8-1.

Table 8-1. Supported PSpice Component Types

COMPONENT:	**GaAsFET**
NAME:	**DGASFET**
REFD:	**B**
PARM 1:	**Model Name**
PARM 2:	**Area Value**
COMPONENT:	**Capacitor**
NAME:	**DCAP,DCAPE**
REFD:	**C**
PARM 1:	**Model Name**
PARM 2:	**Value**
PARM 3:	**Initial Value IC=( )**
COMPONENT:	**Diode**
NAME:	**DDIODE,DZENER**
REFD:	**D**
PARM 1:	**Model Name**
PARM 2:	**Area Value**

COMPONENT:	Voltage-Controlled Voltage Source
NAME:	ESOURCE
REFD:	E
PARM 1:	Gain

COMPONENT:	Current-Controlled Current Source
NAME:	FSOURCE
REFD:	F
PARM 1:	Controlling V Device Name
PARM 2:	Gain

COMPONENT:	Voltage-Controlled Current Source
NAME:	GSOURCE
REFD:	G
PARM 1:	Transconductance

COMPONENT:	Current-Controlled Voltage Source
NAME:	HSOURCE
REFD:	H
PARM 1:	Controlling V Device Name
PARM 2:	Transresistance

COMPONENT:	Independent Current Source
NAME:	DCUR
REFD:	I
PARM 1:	DC Value
PARM 2:	AC Magnitude
PARM 3:	Phase Value
PARM 4:	Transient Specification

COMPONENT:	Junction FET
NAME:	DNJFET,DPJFET
REFD:	J
PARM 1:	Model Name
PARM 2:	Area Value

COMPONENT:	Mutual Coupling Inductor (Transformer)
NAME:	DMUTUAL
REFD:	K
PARM 1:	Inductor Names (separated by spaces)
PARM 2:	Coupling Value
PARM 3:	Model Name
PARM 4:	Size Value

COMPONENT:	Inductor
NAME:	DIND
REFD:	L
PARM 1:	Model Name
PARM 2:	Value
PARM 3:	Initial Value IC=( )

COMPONENT:	MOSFET
NAME:	DNMOSFET,DPMOSFET
REFD:	M
PARM 1:	Bulk/Substrate Node
PARM 2:	Model Name
PARM 3:	Other Parameters

COMPONENT:	Bipolar Transistor
NAME:	DNPN,DPNP
REFD:	Q
PARM 1:	Substrate Node
PARM 2:	Model Name
PARM 3:	Area Value

COMPONENT:	Resistor
NAME:	DRES
REFD:	R
PARM 1:	Model Name
PARM 2:	Value

COMPONENT:	Transmission Line
NAME:	DTRANS
REFD:	T
PARM 1:	Z0 Value
PARM 2:	TD Value
PARM 3:	F Value
PARM 4:	NL Value

COMPONENT:	Independent Voltage Source
NAME:	DVOLT
REFD:	V
PARM 1:	DC Value
PARM 2:	AC Magnitude
PARM 3:	Phase Value
PARM 4:	Transient Specification

COMPONENT:	Subcircuit Call
NAME:	DXSUB
REFD:	X
PARM 1:	Subcircuit Name

COMPONENT:	Ground
NAME:	DSGND
REFD:	None

COMPONENT:	**Notes**
NAME:	**DNOTE**
REFD:	**'.' or '_'**
PARM 1:	**Values and Parameters**

The "IC=( )" Value that appears for capacitors and inductors as parameter 3, needs only the value to be given. CONVERT.EXE will automatically place the "IC=" and the value enclosed in the brackets. If parameter 3 is not defined, then nothing will be output to the PSpice file. The "Z0", "TD", "F" and "NL" values for the Transmission Line part will also be given automatically and functions the same as the "IC=" Value. These are the only components that the value prefix information is supplied for the user. Any other components that use the parameter fields need the information to be entered completely. If the user wishes to leave a parameter blank, this will not cause any problems for the output file.

The **DSGND** object must be used for ground references in PSpice schematics. The standard ground symbol (**GND**) applies the label **GND** to the connection point while DSGND applies label **0** to the connection point.

The **DNOTE** object simply prints out the reference designator followed by the Schematic Editor **DRAW-OBJECT** Parameter String **PARM 1** field entry as a line in the PSpice net list (no node numbers are included). Two exception-case uses of **DNOTE** are used to add non-component (dot) information to the PSpice net list (no reference designators). If a **DNOTE** component reference designator begins with a period (.) or underline (_) character, the component's reference designator will not be included in the PSpice net list and only the **PARM 1** field entry is printed. This feature is used to imbed dot PSpice net list statements within your schematics. **DNOTE** components whose reference designators begin with a period character are placed before

other components in the PSpice net list. **DNOTE** components whose reference designators begin with an underline character are placed after other components in the PSpice net list.

Normally, an alpha or numeric character follows the period or underline character. This orders the PSpice net list statements. To add five statements before the net list component descriptions, use .1, .2, .3, .4, and .5 as the reference designators for the applicable statements.

A sample PSpice schematic, **SPICE.P01**, contains some of the PSpice object types and demonstrates DNOTE use. To create the PSpice net list file (**SPICE.CIR**) from the sample schematic, enter:

SCHPOST SPICE P <Enter> (Creates SPICE.NPL file)
BASICA SPICE <Enter> (Creates SPICE.CIR file)

or

SCHPOST SPICE C <Enter>

Select:

PSpice <Enter> (Creates a SPICE.CIR file).

Compare the SPICE.CIR file with the SPICE.P01 drawing to see the relationship between the SPICE schematic and the SPICE net list.

Using PSpice for Digital/Analog Applications

This conversion differs from the one found in **CONVERT.EXE** with support added for MicroSim's Digital/Analog version.

Before using **PSPCONV.EXE** you will need to run SCHPOST on your schematic and create a pinlist.

EX. **SCHPOST FILENAME P <Enter>**

Then you can execute **PSPCONV** by typing:

PSPCONV FILENAME <Enter>

Give no extension as in SCHPOST

PSPCONV.EXE first reads in the part table **PSPCONV.TBL**. The table contains the name of the part, the first reference designator letter, and the pins and parameters in the order to be output. Next, it opens the input .NPL file and reads in the information for a part, finds a match in the table and outputs the node names in the order the Table specifies. The output file has the extension of **".CIR"**.

The Input Table, **PSPCONV.TBL**, can be modified to add new objects. Use a Non-document text editor. The table is made up of different stages:

NAME - The name is the name of the part. Because many parts have the same pin outs, i.e. 74LS00, 74S00, and 74F00, the wildcard character, '*' is incorporated. The Table entry is, 74*00. This means that when it is trying to make a match from the netlist to the table it will ignore any alpha characters between the digits. Remember the wildcard it only to be used for a sequence of letters with no numbers. The name 7400 is also matched to the 74*00 entry. Some parts are specific or have an 'A' extension. If this is true then you can hardcode these parts in the table. Be sure to enter them in the table before the wildcard entry of the same so the match will not be taken from the wildcard version.

REFD PREFIX - This is the first character of the reference designator that will be expected to match the one found in the netlist. If the match fails a warning is given. The wildcard '*' can be used to suppress this check.

SIZE - This is a number that represents the total number of pins and parameters that should follow. For example, if there are three pins on the part and three possible parameter fields to output the size will be 6.

PINS - These are the pin numbers or letters that represent which net names to output to the .CIR file. The order in which they appear in the table is the order they are to be output. The reason for this is that PSpice wants the pin to appear in a certain order, which may not neccesarily be in numeric order. For example, a 7425 pins are to be output in the following order:

A B C D G Y

The pin numbers of the SCHEMA part are: 1 2 4 5 3 6. Characters (A-Z) can be used for the pins. Only one character per pin.

PARMS - These are optional lines.

1000 - is the Part Name. This is used for the displaying the digital parts name in the output file after the pins.

1001 - is PARM 1. This is used to display the Parameter 1 to the output file.

1002 - is PARM 2. This is used to display the Parameter 2 to the output file.

1003 - is PARM 3. This is used to display the Parameter 3 to the output file.

1004 - is PARM 4. This is used to display the Parameter 4 to the output file.

NOTE: The gate parts have only the 'A' Gate defined. Therefore use only 'A' gates in your design. Then when done with the simulation change your gate sections for the finished schematic and incorporate the full package range, B, C, D, etc. Thus, if a pin in the range of 1 to 20 has no net name found for it a warning will appear indicating a possible problem.

These parts are specific to the Digital PSpice:

DIGITIN - DIGITAL INPUT
DIGITOUT - DIGITAL OUTPUT
VSWITCH - VOLTAGE CONTROLLED SWITCH
ISWITCH - CURRENT CONTROLLED SWITCH

These are explained below.

NOTE: In CONVERT.EXE the "IC=( )" string is output automatically for capacitors and inductors with the value of PARM 3 in the parentheses. This is no longer the case with this conversion program. You will have to type in the full string to the parm of the part in the schematic. This is true for the Z0, TD, F, and NL values of the Transmission Lines.

The CONVERT program of SCHEMA converted the auto-assigned net names, like 01036032 with a value between 1 - 9999. This was to be compatible with U. C. Berkeley SPICE. However, this program is

specific to PSpice versions that can handle a maximum eight character name, thus the net names will retain their original values.

PSpice also uses the first letter of the component's reference designator to determine the part type. Therefore, never use auto-assigned reference designators as they will be incorrect. There are several library objects to be used with PSpice and they are found in the ANALOG.LBR library. In addition to those the following are specific for Digital. They are described in the following Table:

Table Supported PSpice DIGITAL Component Types

COMPONENT: Digital Input
NAME: DIGITIN
REFD: N
PARM 1: Model Name
PARM 2: Digital Net Name, DGTLNET = ( )
PARM 3: Digital IO Model Name
PARM 4: Initial State, IS = ( )

COMPONENT: Digital Output
NAME: DIGITOUT
REFD: O
PARM 1: Model Name
PARM 2: Digital Net Name, DGTLNET = ( )
PARM 3: Digital IO Model Name

COMPONENT: Voltage-Controlled Switch
NAME: VSWITCH
REFD: S
PARM 1: Model Name

COMPONENT: Current-Controlled Switch
REFD: W
PARM 1: Controlling V device Name
PARM 2: Model Name

COMPONENT: Subcircuit Call
NAME: User Definable - Enter the part into the Table
REFD: X

Any other components that use the parameter fields need the information to be entered completely. If the user wishes to leave a parameter blank, this will not cause any problems for the output file.

The DSGND object must be used for ground references in PSpice schematics. The standard ground symbol (GND) applies the label GND to the connection point while DSGND applies label 0 to the connection point.

The DNOTE object works the same as explained earlier in the PSpice for analog simulation.

8.7 Network Capatibility

Using SCHEMA on a network is merely a set of ground rules to which the user must adhere.

If you create two different subdirectories off the \SCHEMA root on the network drive, F:\SCHEMA\SMITH and F:\SCHEMA\JONES. Place in each of these two subdirectories a SCHEMA.CFG and a

SCHEMA.MGR specific to these two individuals workstation hardware and manager preference.

The OMATION environment variable will need to be set up for each workstation. OMATION=F:\SCHEMA\SMITH and OMATION=F:\SCHEMA\JONES. There should be no SCHEMA.CFG or SCHEMA.MGR files in the SCHEMA directory to be confused with these user ones. The user should have SCHEMA in his PATH= statement of his AUTOEXEC.BAT for the .BAT and .EXE files. The user can then type SCHEMA from any directory at his worstation, his manager and config file are loaded from his user specific subdirectory. If the two users are reading from the same library, and one user creates a part and tries to save it, he gets the message, "DEVICE ERROR: Enter, Retry, Ignore, Abort". If one of the users leaves the drawing releasing the library and the other user presses 'R' to retry the part is saved. If two users load the same schematic page and one alters and saves the file. When the other trys to save a warning is displayed stating that the schematic page changed since it was loaded on his machine. These functions have been tested on Omation's NOVELL PC network.

Other issues are determined by the network supervisor. He may wish to make all the files in the SCHEMA directory Read only. Be sure *NOT* to make the S$$$.BAT Read only since it is a batch file the manager creates so that it can return back to itself after a command is batched. He may want to restrict users from writting to other user directories. He may wish to give the user only Read privaledges to other user's subdirectories. He may want to make the libraries read only except for one user specified to alter them.

Appendix A. Installing SCHEMA III

A.1 INTRODUCTION

This appendix describes how to install SCHEMA III system files, libraries and hardware device drivers.

The INSTALL.EXE program included in PROGRAM DISK 1 allows you to install SCHEMA III files on the drive and DOS sub-directory of your choice.

Advanced DOS users may wish to install SCHEMA III manually. *Do not attempt to manually install SCHEMA III unless you are familiar with DOS.*

The SCHEMA III Program disks have been archived to save disk space. A program called PKUNZIP.EXE is used to unarchive the pertinent SCHEMA files for use with SCHEMA III. The unarchive process is described later.

A.2 SCHEMA III DISK CONTENTS

The SCHEMA III Program disks contain all of SCHEMA III's program and drawing files archived into seven files: SCHEMA1.ZIP, SCHEMA2.ZIP, SCHPCB.ZIP, DRAWINGS.ZIP, DRIVERS.ZIP, and NETLIST.ZIP. The SCHEMA III schematic capture system files are archived in two separate files, SCHEMA1.ZIP and SCHEMA2.ZIP, due to the volume and size of SCHEMA III program files. 3.5 Disk distribution contains 3 disks. One Program and 2 Libraries.

DRAWINGS.ZIP contains sample and tutorial schematics. DRIVERS.ZIP contains the SCHEMA III printer driver files. LIBRARY.ZIP contains the tutorial library LOGIC.LBR. NETLIST.ZIP contains the net and pin list conversion programs. SCHPCB.ZIP contains additional system files necessary for SCHEMA-PCB users.

The additional disks contain the SCHEMA III libraries. The partitioning of the SCHEMA III archive files for distribution on 1.2 Meg diskettes is illustrated in table A-1.

Files are unarchived to their original form by running the INSTALL program or manually using the program, PKUNZIP.EXE, distributed on program disk 1. For manual installation, PKUNZIP.EXE is used as follows:

Copy **PKUNZIP.EXE** from program disk 1 to the current working directory.

To unarchive all files from the SCHEMA1.ZIP file on program disk 1 in drive A: to the C:\SCHEMA directory type:

PKUNZIP A:\SCHEMA1 C:\SCHEMA -O

To unarchive only .EXE files from SCHEMA1.ZIP to the C:\SCHEMA directory type:

PKUNZIP A:\SCHEMA1 *.EXE C:\SCHEMA -O

To unarchive a single file from SCHEMA1.ZIP type:

PKUNZIP A:\SCHEMA1 SCHCFG.EXE C:\SCHEMA -O

The /R option at the end of the command line tells **PKUNZIP.EXE** to overwrite files of the same name without prompting for permission first.

A.3 MANUALLY INSTALLING SCHEMA III TO ON HARD DISK SYSTEMS

Former SCHEMA and SCHEMA II users should take steps to salvage or protect custom parts created and saved to existing libraries prior to installing the new libraries.

There are many different ways to manually install SCHEMA III on a hard disk system. The method described herein is just one method. Feel free to organize the SCHEMA III files/directories differently to suit individual preferences.

a. Create a new sub-directory (typically C:\SCHEMA).

b. Unarchive all files from the SCHEMA III Program disks' SCHEMA1.ZIP, SCHEMA2.ZIP, DRAWINGS.ZIP and LIBRARY.ZIP files to the newly-created sub-directory. SCHEMA-PCB users must also unarchive the SCHPCB.ZIP file to this sub-directory. *SCHEMA III and SCHEMA-PCB must be installed on the same drive.*

c. Unarchive the desired printer driver file from the SCHEMA III Program disks' DRIVERS.ZIP to the newly-created sub-directory (e.g., **PKUNZIP A:\DRIVERS EPSON80.DRV C:\SCHEMA -O**).

d. Unarchive all files from the SCHEMA III Library disks to the newly-created sub-directory.

The TTL library is distributed on two separate diskettes as TTL1 and TTL2. After unarchiving the libraries may be combined into one large library, if desired, by creating a new large library using SCHLIB and copying both into the new one. See Chapter 6, *SCHLIB*, for details.

e. Modify the AUTOEXEC.BAT file in the root directory. Add the system files' directory (C:\SCHEMA) to the **PATH** variable. Create an environment variable named OMATION calling out the system files' directory by adding the line:

SET OMATION=C:\SCHEMA

NOTE

The system files, printer driver(s) and libraries may reside in different directories. Be sure to specify full paths for each of these in the SCHEMA.CFG and <Drawing>.CFG files. Refer to Chapter 5, SCHCFG, for details.

f. Modify the CONFIG.SYS file in the root directory to contain the **FILES=n** line, where **n** is at least **20** (SCHEMA III and SCHEMA-PCB require a minimum **n** value of **20**). The **FILES=n** line sets the maximum number of open files allowed. If you are running out of environment space place this statement: **SHELL=C:\COMMAND.COM /e:255 C:\ /P**

g. If you are using a mouse, you must install the mouse driver. SCHEMA III communicates with the mouse through a Microsoft-compatible mouse driver. This is one of the drivers supplied with your mouse. Install the mouse driver using a DOS CONFIG.SYS entry, an entry in the AUTOEXEC.BAT file, or through a stand-alone command. Refer to your mouse owner's manual for additional information on the mouse driver.

h. Reboot system to effect new AUTOEXEC.BAT and CONFIG.SYS.

i. SCHDRAW uses the DOS-supplied PRINT.COM program to print files in background. The PRINT.COM program must be initialized prior to invoking SCHDRAW. To initialize PRINT.COM enter: **PRINT <Enter>** (where **<Enter>** is a single keypress of the **<Enter>** key). After loading, PRINT prompts for the name of the list device (normally press **<Enter>** for the default). If desired, "**PRINT**" may be placed in the AUTOEXEC.BAT file.

j. Configure the SCHEMA manager to support SCHEMA III by appending the SCHEMA III manager template DEFSCH.MGR to the default manager configuration file, SCHEMA.MGR, and editing the header and paths information as required with an ASCII text editor. SCHEMA-PCB users may want to append the DEFPCB.MGR template to SCHEMA.MGR as well. See the DOS **COPY** command in your DOS manual for instructions on appending text files. See Appendix G, *SCHEMA Manager Configuration*, for information on manager configuration syntax and editing the manager configuration file.

A.4 CONFIGURING SCHEMA III TO WORK WITH YOUR HARDWARE

All SCHEMA III default configuration parameters are contained in the SCHEMA.CFG file, archived in SCHEMA1.ZIP of the SCHEMA III Program disk 1. We will now modify this SCHEMA.CFG file to match your exact hardware configuration.

NOTE

*All keyboard entries (and screen displays) appear as boldfaced text in this manual. Also, **<Enter>** refers to a single keypress of the Enter key (the*

same is true of <Esc>, <Backspace>, etc.). Refer to Appendix C, Text Editing Keyboard Usage, for a complete listing of editing keys used with SCHEMA III.

If, during operation, SCHEMA III does not respond as stated, refer to Appendix D, *Error Messages*.

Do one of the following procedures that applies to your system disk configuration :

a. HARD DISK SYSTEMS

Ensure the installation instructions in the preceding paragraphs have been followed. Next, boot DOS. Now make the directory that contains the SCHEMA III system files (typically C:\SCHEMA) the current directory.

b. ALL SYSTEMS

To modify the SCHEMA.CFG file, we need to invoke SCHEMA III's Configuration Manager program, SCHCFG. SCHCFG is discussed in detail in Chapter 5. To invoke SCHCFG, after the DOS prompt, enter:

SCHCFG <Enter>

After a few seconds, SCHCFG loads with the current SCHEMA.CFG file and displays the SCHCFG Main Configuration Menu, shown in figure A-1. This menu lets you configure SCHEMA III to match your PC's hardware configuration. To select each variable, highlight the variable using the cursor keys or mouse followed by pressing **<Enter>** or the mouse button.

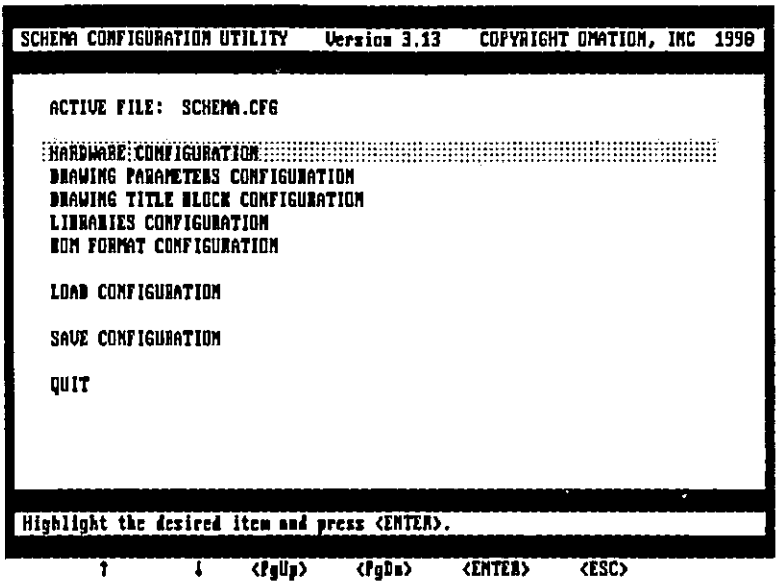


Figure A-1. SCHCFG Main Configuration Menu

1. Highlight **HARDWARE CONFIGURATION** and press **<Enter>**. Highlight **GRAPHICS ADAPTER SELECTION** and press **<Enter>**. Select the **GRAPH** variable to match your graphics adapter type and press **<Enter>**. Refer to the **GRAPH** command in Chapter 5, *SCHCFG* for screen color settings.

2. Highlight **PRINTER DRIVER SELECTION** and press **<Enter>**. Select the **PRINT** variable to match your printer type and press **<Enter>**. The **PRINT** field contains the name and location of the printer driver. Thus, the **PRINT** field must also contain the printer driver's disk/directory, name, and extension (always .DRV). The printer drivers are generally located with the SCHEMA III programs in the

directory that contains the SCHEMA III programs.

NOTE

*A printer must be assigned to the **PRINT** variable even if a printer is not used.*

3. If you are using a plotter, highlight **PLOTTER OPTIONS SELECTION** and press **<Enter>**. Select the plotter variable to match your plotter type. Unlike **PRINT**, **PLOTTER** does not contain the driver location (plotter drivers are part of the SCHPLOT program). Select **PXLEN, PYLEN, PXOFF, PYOFF and PENW** ..... and press **<Enter>**. Select **EXIT** to return to the Hardware Configuration Menu.

Highlight **DISPLAY/EDIT CURRENT SETTINGS** and press **<Enter>**. Confirm the configuration variable settings and their respective paths. If any changes need to be made, highlight the variable, edit the existing setting and press **<Enter>**.

4. Select **EXIT** twice to return to the Main Configuration Menu.

5. The Libraries Configuration Menu assigns object libraries to your drawing. Each entry must contain the disk/directory and complete library name with .LBR extension. if applicable, is a part of each library entry. For example, if TTL1.LBR is in the C:\SCHEMA sub-directory, enter:

C:\SCHEMA\TTL1.LBR <Enter>

7. Select **EXIT** to return to the Main Configuration Menu.

8. This completes our session. Now select **SAVE** followed by **<Enter>** to save the SCHEMA.CFG file to the current directory. Press **<Enter>** to overwrite the existing SCHEMA.CFG file. Select **QUIT** followed by **<Enter>** to return to DOS.

A.6 SCHEMA III FILE CONVENTIONS

SCHEMA III files are organized according to function. Table A-1 lists the files included with SCHEMA III and includes a brief description of each file.

There are five major file types:

- *Executable.* Executable files have a .EXE extension. These files are the only executable programs included with SCHEMA III (see table A-1). All other files are data files.

- *Library.* Library files have a .LBR extension. These files contain objects used in drawings. Each library contains many objects, yet appears as a single file to DOS. Only SCHEMA III executable files have the capability to use objects from library files.

- *Configuration.* Configuration files are used by SCHEMA III's executable programs for configuration and drawing parameter purposes. These files are standard text files and can be read from and written to with any text editor (such as DOS EDLIN), but are normally modified with the SCHCFG program. There is one SCHEMA.CFG file for all of SCHEMA III's programs and generally one DRAWING.CFG file for each schematic drawing.

- *Schematic Drawing.* Schematic drawing files have extensions of .Pxx, .Bxx, and .Txx, where xx is a numeric representation of the page number. Files ending with .Pxx are current versions, .Bxx files are backup versions, and .Txx files are temporary versions (such as print files) of the applicable schematic drawings. A typical multiple-page schematic consists of several files. Each file has an identical file name with a unique file extension. For example, if a schematic is named MEMORY, file MEMORY.P01 contains the first drawing page, MEMORY.P02

contains the second page, etc. Backup (.Bxx) and temporary (.Txx) files are named in a similar manner. SCHEMA III is shipped with several predefined drawing pages including blank pages with borders and completely blank pages. These pages have .P99 extensions.

- *Driver.* Driver files have .DRV extensions. These files contain hardware drivers for the various printers that SCHEMA III supports. These files are loaded by the executable files after the executable files have initially loaded.

- *Other.* Additional file types are created when net lists are generated by SCHEMA III. Net lists are assigned the name of the applicable schematic drawing, followed by a .NET extension. Pin lists are similar to net lists except pin lists have .NPL extensions. SPICE net list files have .CIR extensions. The actual net and pin list conversion programs included with SCHEMA III are written in BASIC and thus, have file extensions of .BAS. SCHBACK ASCII 'was/is' files are assigned the name of the applicable schematic drawing followed by a .REF extension. Regions that are defined and saved in SCHDRAW have file extensions of .RGN.

The file READ.ME is included on the SCHEMA III Program disk and should be read. READ.ME contains important, updated user information and references additional files.

Table A-2 contains a summary of SCHEMA III file extensions.

Table A-1. SCHEMA III Disk Distribution

TYPE	FILENAME	DESCRIPTION
Executable		
	BACK.EXE	PCB Back Annotator
	CONVERT.EXE	Net/Pin List Conversion Program
	INSTALL.EXE	Installation Program
	SCHBACK.EXE	PCB Back Annotator
	SCHCFG.EXE	Configuration File Manager
	SCHDRAW.EXE	Drawing Editor
	S$.EXE	SCHEMA Manager
	SCHLIB.EXE	Library Manager
	SCHLOCAL.EXE	Library Archiver
	SCHPAGE.EXE	Page Maker
	SCHPLOT.EXE	Foreground Plotter Routine
	SCHPOST.EXE	Post-Processor
	POST$$1.EXE	Post-Processor
	POST$$2.EXE	Post-Processor
	POST$$3.EXE	Post-Processor
	POST$$4.EXE	Post-Processor
	POST$$5.EXE	Post-Processor

	PSPCONV.EXE	PSpice Digital Conversion
	SCHPRINT.EXE	Foreground Printer Routine
Batch		
	SCHEMA.BAT	SCHEMA Batch File
Configuration		
	SCHEMA.CFG	Configuration File
Other		
	READ.ME	Information Not in Manual
	TUTORIAL.REF	SCHBACK ASCII 'Was/Is' File
Schematic Drawing		
	PPPAGE1.P99	Portrait C-size Blank Page
	PPPAGE2.P99	Portrait D-size Blank Page
	PPPAGE3.P99	Portrait E-size Blank
	PPAGE1.P99	Virtual C-size Blank Page
	BPAGE1.P99	Virtual C-size Blank Page with Border
	PPAGE2.P99	Virtual D-size Blank Page
	BPAGE2.P99	Virtual D-size Blank Page with Border

	PPAGE3.P99	Virtual E-size Blank Page
	BPAGE3.P99	Virtual E-size Blank Page with Border
	PPAGEA.P99	A-size Blank Page
	PPAGEB.P99	B-size Blank Page
	PPAGEC.P99	C-size Blank Page
	PPAGED.P99	D-size Blank Page
	PPAGEE.P99	E-size Blank Page
Schematic Drawing		
	BPAGEA.P99	A-size Bordered Page
	BPAGEB.P99	B-size Bordered Page
	BPAGEC.P99	C-size Bordered Page
	BPAGED.P99	D-size Bordered Page
	BPAGEE.P99	E-size Bordered Page
	PPPAGEA.P99	Portrait A-size Page
	PPPAGEB.P99	Portrait B-size Page
	PPPAGEC.P99	Portrait C-size Page
	PPPAGED.P99	Portrait D-size Page
	PPPAGEE.P99	Portrait E-size Page
	SPICE.P01	Sample Schematic
	TUTORIAL.P01	Sample Schematic
	TUTORIAL.P02	Sample Schematic
Driver		
	IMAGE15.DRV	Apple Imagewriter 15" Printer Driver
	IMAGE8.DRV	Apple Imagewriter 8" Printer Driver
	ICANON.DRV	Canon LBP-8 A2

	Laser Printer Driver
CI1550.DRV	C-Itoh 15.5" Printer Driver
CI8510.DRV	C-Itoh 10" Printer Driver
CITMSP15.DRV	Citizen MSP 15" Printer Driver
CITMSP10.DRV	Citizen MSP 10" Printer Driver
EPSON100.DRV	Epson 15" Printer Driver
EPSON80.DRV	Epson 10" Printer Driver
ILQ1500.DRV	Epson LQ-Series 15" Printer Driver
LQ2500.DRV	Epson LQ-Series 2500 Printer Driver
GENICOM.DRV	Genicom 3000 Series Printer Driver
DESKJET.DRV	HP Deskjet Printer Driver
HPTJET1.DRV	HP PaintJet Printer Driver
IHPLAS.DRV	HP LaserJet+ Printer Driver
IBMCOLOR.DRV	IBM Color Printer Printer Driver
NULL.DRV	Null Printer Driver
OKI82A.DRV	Okidata 82A Printer Driver
OKI83A.DRV	Okidata 83A

	Printer Driver
IOKI193.DRV	Okidata 193 15" Printer Driver
IOKI192.DRV	Okidata 192 10" Printer Driver
OKI2410W.DRV	Okidata 2410 15" Printer Driver
OKI2410N.DRV	Okidata 2410 10" Printer Driver
OKIML92.DRV	Okidata ML92 Printer Driver
OKIML93.DRV	Okidata ML93 Printer Driver
PCX.DRV	PC Paintbrush Printer Driver
PTRON600.DRV	Printronix 300/600 Printer Driver
PRO100.DRV	Proprinter Printer Driver
PRO80.DRV	Proprinter Printer Driver
STAR80.DRV	Star 10" Printer Driver
STAR100.DRV	Star 15" Printer Driver
TI850.DRV	Texas Instruments 850 Printer Driver
TIFF.DRV	TIFF Printer Driver
ITOS351.DRV	Toshiba 350/1351 15" Printer Driver
ITOS340.DRV	Toshiba 340/1340 10"

Printer Driver

Netlist Conversions

CADNETIX.BAS	Cadnetix/ SCI-CARDS Net List Program
CALAY.BAS	Calay Net List Program
CBDS.BAS	CBDS Net List Program
CV.BAS	Computervision Net List Program
DATACON.BAS	dataCon Wire-Wrap Net List Program
EDIF.BAS	EDIF Net List Program
EE.BAS	Visionics Net List Program
FNNET.BAS	FutureNet Net List Program
FNPIN.BAS	FutureNet Pin List Program
IG.BAS	Intergraph Net List Program
PDIF.BAS	P-CAD PDIF Net List Program
RR.BAS	Racal-Redac Net List Program
SPICE.BAS	SPICE Pin List Program
TELESIS.BAS	Telesis Net List Program

PCB Files:

Batch		
	SCHPCB.BAT	SCHEMA-PCB Batch File
Configuration		
	PCB.CFG	PCB Tutorial Configuration File
	PCBECO.CFG	PCBECO Tutorial Configuration File
Executable		
	PCBACK.EXE	Back Annotator from PCB
	RETDIR.EXE	SCHEMA.BAT Utility File
	SCHDIR.EXE	SCHEMA.BAT Utility File
	SCHECO.EXE	Foward and Back Annotator
	SCHEM.EXE	SCHEMA.BAT Utility File
	SCHFWD.EXE	Forward Annotator to PCB
Library		
	PCB.LBR	PCB Tutorial Library
Schematic Drawing		
	PCB.P01	PCB Tutorial Drawing

	PCB.P02	PCB Tutorial Drawing
Library		
CMOS LIBRARY	CMOS.LBR	CMOS Library
ECL LIBRARY	ECL.LBR	ECL Library
IEEE/IEC LIBRARY	IEEE.LBR	IEEE/IEC Component Library
MEMORY LIBRARY	MEMORY.LBR	Memory Library
ANALOG LIBRARY	ANALOG.LBR	Analog Library
MICRO-PROCESSOR AND GRAPHICS	MICRO.LBR	Microprocessor Library
	GRAPH.LBR	Graphics Object Library
TTL LIBRARY 1	TTL1.LBR	TTL Library
TTL LIBRARY 2	TTL2.LBR	TTL Library

Table A-2. SCHEMA III File Extension Summary

EXTENSION	DESCRIPTION
.BAS	BASIC Program Files
.Bxx	Backup Schematic Drawing Files
.CFG	SCHEMA III Configuration Files
.CIR	SPICE Net List Files
.DBF	DBASE BOM Output Files
.DRV	Hardware Driver Files
.ECO	SCHECO Forward Annotation Files
.EXE	Standard DOS Executable Files
.FWR	SCHECO Current Database Files/ SCHPOST PCB Pin List Option
.GBL	SCHPOST - Hierarchical Global net file
.HBK	SCHBACK Hierarchical Files

.HSF	SCHPOST Hierarchical Prefix Files
.LBK	SCHLIB Library Backup
.LBL	SCHPOST - label swap file
.LBR	Library Files
.NET	Net List Files
.NPL	Pin List Files
.NST	Post-Processor netlist
.Oxx	Object file for schematic page (pinlist)
.PST	Post-Processor parts list
.Pxx	Schematic Drawing Files
.P99	Blank Drawing Page Files
.REF	SCHBACK ASCII 'Was/Is' Files
.RGN	Region Files
.SCH	SCHEMA-PCB Pin List Files

.SDF	Standard Delimited Output Files
.Sxx	SCHPLOT File Output Option Files
.TXT	Delimited BOM Output Files
.Txx	Temporary Schematic Drawing Files
.WKS	LOTUS BOM Output Files
.$$$	SCHEMA-PCB Board Database Netlist Files

where: xx = 01 to 98

<Left Blank>

Appendix B. Net/Pin List Formats

B.1 INTRODUCTION

The SCHPOST net and pin list outputs allow interfacing a SCHEMA III schematic to other popular CAE, PCB layout, simulation, etc. systems.

The net list is a machine-readable file (compared to the SCHPOST wire list output, which is considered human-readable). Both the net list and wire list contain the same connection information. However, the connection information is formatted differently in each list. In addition, the net list contains a limited amount of component-type information. The wire list contains headings, page breaks, and easily-read net divisions. The net list is an ASCII file of component types and nets with no headings, page breaks, etc. The net list file is used as the input to a user-written program. This program takes the net list input and reformats the information into a form compatible with another computer system.

The pin list is a superset of the net list, containing more component and net section data.

B.2 NET LIST FORMAT

The net list format (figure B-1) is straightforward. It contains only two sets of information: a set of component descriptions and a set of nets. The net list file is a standard ASCII file. Each line is terminated with a carriage return (0DH) and line feed (0AH). Each component description is comprised of multiple lines. The first line contains an

open bracket character ([). The next line contains the component name, including sub-type and alternate orientation identifiers. The third line contains the component's reference designator.

```
Component Section, one entry for each
component ( sorted be REFD field )

      [
      NAME
      REFD
      DESC1
      DESC2
      DESC3
      DESC4
      ]

Net Section, one entry for each net

      (
      LABEL
      REFD-PIN
      .
      .
      REFD-PIN
      )

Defintions:
NAME -  Name Object name of the component
REFD - Component's reference designator
DESC1-4 - Description 1 thru 4 fields
LABEL - Label name
PIN - Pin name or number
LABEL - Label name
```

Figure B-1. SCHEMA III Net List Format

The next four lines contain the component's DESC 1 - DESC 4 fields. If the DESC fields contain any parameter replacements (<P1>, <P2>, <P3>, or <P4>), the replacements are automatically done. The last line of the component description contains a close bracket character (]). Nets follow the component descriptions. A component description exists for each component and/or component section. For example, a four-section 74LS00 has four unique component descriptions.

Each net is comprised of multiple lines. The first line contains an open parenthesis character ((). The next line contains the net label. As with the wire list, if a net does not have a label, one is automatically created. A net name is assigned by generating an eight digit number. The first two are the page number, the next three are the X coordinate and the last three are the Y coordinate. The next lines of each net are net components. All nets have at least two components. The last line of all nets contains a single close parenthesis ()). If another net follows, an open parenthesis follows the closed parenthesis. The last net in the file is followed by an end-of-file character (01AH). This is the last character in the file.

B.3 PIN LIST FORMAT

The pin list format (figure B-2) also contains two sets of information: an expanded set of component descriptions and nets. The pin list file is a standard ASCII file. Each line is terminated with a carriage return (0DH) and line feed (0AH). Each component description is comprised of multiple lines.

The first component section line contains an open bracket character ([).

The second line contains the component's Name Object name.

```
Component Section, one entry for each
component ( sort by REFD )

      [
      NAME
      REFDNUM,REFD,PAGE
      DESC1
      DESC2
      DESC3
      DESC4
      P1
      P2
      P3
      P4
      PIN,ATTR,LABEL,LABELNUMBER
      .
      .
      PIN,ATTR,LABEL,LABELNUMBER
      ]
( PIN lines sorted by PIN field )

Net Section, one entry for each net

      (
      LABEL,LABELNUM
      REFDNUM,REFD,ATTR,PIN,PAGE
      .
      .
      REFDNUM,REFD,ATTR,PIN,PAGE
      )

Definitions:
NAME - Object Name of the component
REFD - Component's reference designator
REFDNUM - Number assigned to each reference
designator
```

```
PAGE - The drawing page number where
component appears
DESC1-4 - Description 1 thru 4 fields
P1-4 - Parameters 1-4 fields from Parameter
String
PIN - Pin name or number
ATTR - Pin type ( O,I,T,X, @, etc )
LABEL - LAbel name
LABELNUM - Number assigned to each net (
label )
```

Figure B-2. SCHEMA III Pin List Format

The third line contains the component's reference designator number, reference designator, and schematic page number the component appears on. The reference designator number is assigned to components on an incrementing basis.

For example, the first component is 1, the next component is 2, etc.

The fourth through seventh lines contain the **DESC 1** through **DESC 4** fields, respectively.

The eighth through eleventh lines contain the component's Schematic Editor **DRAW-OBJECT** Parameter String **PARM 1-4** entries, respectively.

The twelfth and remaining component section lines contain sorted lists of each component's pin number (name), definition (I, O, T, etc.), signal label name, and label number. The label number is a number assigned to each net (label), identical to the net number in SCHEMA's wire list

output. Pins that are defined as power and ground have pin definitions of @, not I.

The last component section line contains a close bracket character (]). The net section follows the component section.

Each net is comprised of multiple lines.

The first net section line contains an open parenthesis character (().

The second line contains the net label and label number. Net numbers are assigned sequentially and are identical to the wire list net numbers. As with the SCHEMA III wire list output, if a net does not have a label, one is automatically created. Created labels consist of an eight digit number as discussed in paragraph B.2.
The third and remaining lines of each net section contain the list of components in the net. Each line contains the component reference designator number, reference designator, pin definition, pin number (name), and the schematic page number on which the component appears.

The last line of all nets contains a single close parenthesis ()). If another net follows, an open parenthesis follows the closed parenthesis. The last net in the file is followed by an end-of-file character (01AH). This is the last character in the file.

B.4 NET/PIN LIST CONVERSION PROGRAMS

The net/pin list conversion programs included with SCHEMA III are:

CBDS Format IBM Mainframe
CADNETIX / SCI-CARDs
CALAY
Computer Vision
dataCON Wire-Wrapping
EDIF 1 0 0 Netlist
EDIF 2 0 0 Netlist
EDIF 2 0 0 Flat for PlusLogic
Visionics EE Designer
FutureNet Netlist
FutureNet Pinlist
Intergraph
PCAD-PDIF
Racal Redac
Redac Interface Format (RINF)
Redac CADSTAR Pro Body/Component List
PSpice
Telesis

The CONVERT.EXE program is an executable file and cannot be modified. There are equivalent BASIC source code programs archived in the file NETLIST.ZIP on the distribution disks. See Chapter 7, *SCHPOST* and Chapter 8, *Utilities*, contains additional information on the net/pin list conversion programs.

<Left Blank>

Appendix C. Keyboard Usage

C.1 INTRODUCTION

This appendix describes SCHEMA III's use of the keyboard function keys. Editing keys not described in this appendix are not used by SCHEMA III.

C.2 TEXT EDITING KEYS

The Manager, SCHDRAW, SCHCFG, and SCHLIB use a common set of keyboard text editing keys. The following keys are the only editing keys recognized by these programs when entering text in the Text Window:

<Esc> - Clears the entire text line and moves the cursor to the first position of the line.

<Backspace> - Erases the character one position to the left of the cursor and moves the cursor back one space. If the cursor is not at the end of the line, all characters to the right of the cursor are moved left one position.

<Del> - Erases the character at the current cursor position. If the cursor is not at the end of the line, all characters to the right of the cursor are moved left one position.

<Left Arrow> - Moves the cursor left one position.

<Right Arrow> - Moves the cursor right one position.

<Home> - Moves the cursor to the first character position.

<End> - Moves the cursor to the last character position.

When the cursor is in the middle of a line (characters to right of cursor), characters are inserted one position at a time, to the right of the cursor.

C.3 SCHDRAW FUNCTION KEYS

SCHDRAW uses the following set of function keys when in the Schematic and Object Editor (when the Text Window is not active).

<Up Arrow> - Moves the cursor up

<Down Arrow> - Moves the cursor down

<Left Arrow> - Moves the cursor left

<Right Arrow> - Moves the cursor right

<Ctrl><Left Arrow> - Jumps the cursor to the left edge of the Drawing Window

<Ctrl><Right Arrow> - Jumps the cursor to the right edge of the Drawing Window

<PgDn> - Jumps the cursor to the bottom edge of the Drawing Window

<PgUp> - Jumps the cursor to the top edge of the Drawing Window

<Home> - Jumps the cursor to the center of the Drawing Window

<F1> - Next keypress is the first letter of the desired Primary Command

<F2> - Next keypress is the first letter of the Secondary Command

<F3> - Redisplays the last text string entered in the Text Window

<F4> - Turns the **CONFIG-DIAG** command on or off

<F5> - Turns the Cursor Coordinate Window on or off

<F6> - Redraws the Drawing Window

<F7> - ZOOM In one level

<F8> - ZOOM Out one level

<F9> - Repeats the previous command

<F10> - Turns the Cursor Grid (**C GRID**) on or off

<Ins> - Simulates pressing the left mouse button to select a command or do an editing function

<Esc> - Activates the **UNDO** command

C.4 PARTS BROWSING FUNCTION KEYS

The following keys perform these functions only when using the parts browsing feature:

<Home> - Moves the cursor to the first name on the screen.

<PgUp> - Moves the cursor to the previous screen.

<PgDn> - Advances the cursor to the next screen.

<End> - Moves the cursor to the last name on the screen.

<Left Arrow> - Advances the cursor to the previous name on the current screen.

<Right Arrow> - Moves the cursor to the next name on the current screen.

<Up Arrow> - Moves the cursor to the name on the previous row, but remains in the same column.

<Down Arrow> - Moves the cursor to the name on the next row, but remains in the same column.

C.5 INTERNATIONAL CHARACTERS

The following international characters can be generated on schematic drawings anywhere text is entered by pressing and holding the <Alt> key down while entering the corresponding number on the numeric keypad.

Ç	128	ë	137	ő	148	ò	162
ü	129	é	138	ó	149	ù	163
è	130	ï	139	û	150	n	164
â	131	î	140	ú	151	N	165
ä	132	ì	141	ÿ	152	u	230
á	133	A	142	O	153	Ω	234
a	134	A	143	U	154		
ç	135	E	144	å	160		
ê	136	ô	147	í	16		

<Left Blank>

Appendix D. Error Messages

D.1 INTRODUCTION

This appendix contains a listing of all error messages generated by SCHEMA III. Numbers enclosed in parentheses immediately following the listings refer to the program(s) with which the message is associated:

SCHDRAW (1)
SCHPLOT (2)
SCHPRINT (3)
SCHPOST (4)
SCHLIB (5)
SCHCFG (6)
SCHBACK (7)
INSTALL (8)
MANAGER (9)

NOTE

If an error occurs in SCHDRAW and the Error Window is displayed, press any key to acknowledge the error, remove the Error Window, and resume operation.

ABOVE FILE NOT FOUND....ABORT - The file requested was not found. (1)

BIT IMAGE TOO LARGE, PLACE AND RE-EDIT - The bit image is too large to bring into the bit editor. (1)

BODY NAME TOO LONG - The BODY field (NAME-BODY) placed in the Name Object can only be a prefix of *five* characters. e.g. If the BODY Object was, LATCH-1A then the BODY Field in the Name Object could be at the most, LATCH. (1)

BODY OBJECT REFERENCES ITSELF - It is illegal for a body object to contain a name object that references that same body object. See Chapters 1 and 3 for the proper procedure when creating a part. (4)

BOM ERROR: DESCRIPTION NOT FOUND - A description field, specified in the BOM3 entry of the .CFG file, was not found. (4)

BOM ERROR: PARAMETER USED IN DESCRIPTION NOT FOUND - A parameter field that was used in a name object description, specified in the BOM3 entry of the .CFG file, was not found. (4)

BOM WARNING: COMBINE FIELD PARAMETERS ARE NOT ON FIELD BOUNDARIES - The combine field parameters, in the BOM3 entry of the .CFG file, are not on field boundaries. The first parameter must be the position of the first character of a field and the second parameter must be the position of the last character of a field. If the parameters are not on field boundaries, then they will automatically be rounded off to the nearest field boundaries. The first parameter will be rounded to the next lower first character position and the second parameter will be rounded to the next higher last character position. (4)

CANNOT FIND LIBRARY LIBR_NAME - The library, *libr_name*, specified in the configuration file cannot be opened. The library does not exist, the path is incorrect, or too many files are open. The number of files allowed to be open can be increased by changing the **FILES=** statement in your CONFIG.SYS file. (2)

CANNOT OPEN - This indicates that either a library or a drawing file is either unavailable to be opened or restricted.

CONFIGURATION FILE "SCHEMA.CFG" NOT PRESENT...ABORT - The configuration file, SCHEMA.CFG, can not be found. The SCHEMA.CFG configuration file must be present in the current directory/disk or in the directory specified by the OMATION evironment variable. Either the drawing directory/current disk is invalid or the SCHEMA.CFG file has not been placed in the drawing directory. (1,3)

DEVICE ERROR: ENTER Retry, Ignore, Abort - A DOS device error has occurred. Enter: R, I, or A. The device is either a disk or the printer. Retry retries the device (hopefully, after the problem is corrected). Ignore clears the data and proceeds (probably never entered). Abort exits the program and returns to DOS. If using a network, e.g. you may be saving a library part to a library some else has currently opened with read/write access. (1,8)

DOT CONNECTING OVERLAYED WIRES ON PAGE: DRAWING.XXX LOC: X,X-X,X - This message indicates that a dot is connecting a wire which is overlayed with a duplicate wire. (4)

ERROR HELP FILE NOT FOUND - The HELP file, SCHEMA.HLP defualts to the OMATION environment variable first if it exists then the current directory. Check that the OMATION enironment varible points to the correct path to find the SCHEMA.HLP. (1)

ERROR IN OPENING PRINT OUTPUT - The print device (file or printer) could not be opened. The disk is full or the printer is malfunctioning. (1,3)

ERROR INSUFFICIENT DISK SPACE - The number of bytes actually written during a file copy was less than the number of bytes requested and probably indicates a full disk or drive. **(8)**

ERROR OPENING DEVICE COMN - ABORT. - The device COMn could not be opened. Either the device does not exist or too many files are open. The number of files allowed to be opened can be increased by changing the **FILES=** statement in your CONFIG.SYS file. **(2)**

ERROR OPENING OUTPUT FILE FILENAME - ABORT. - The file, *filename*, could not be opened. Either the path is invalid or too many files are open. The number of files allowed to be open can be increased by changing the **FILES=** statement in your CONFIG.SYS file. **(2)**

ERROR READING CONFIGURATION FILE...ABORT - In reading the associated configuration file, an error occurred. The file or disk is probably bad. **(1,3)**

ERROR READING OBJ_NAME FROM LIBRARY LIBR_NAME - The object *obj_name* cannot be read from the library *libr_name*. The library file or disk is bad. **(2)**

ERROR READING PRINT HANDLER - An error has occurred in reading the print handler (**.DRV** file). The file or disk is bad. **(1,3)**

ERROR WRITING OUTPUT - ABORT. - An error occured writing data to the output device/file. **(2)**

ERROR: %S FILE IS MISSING OR BAD - The **.MGR** manager configuration file is not present in the current directory or in the SCHEMA system files directory defined by the environment variable OMATION (typically C:\SCHEMA) or the OMATION environment

variable does not exist in the AUTOEXEC.BAT file in the root directory. (9)

ERROR: %S SYNTAX ERRORS. - The .MGR manager configuration file syntax is incorrect. (9)

ERROR: CANNOT INSTALL OR OPERATE WITH SINGLE DRIVE! - The install was attempted to and from the same drive. The "install from" drive must be different from the "install to" drive. (8)

ERROR: CANNOT UNARCH FILES - The unarchiver ARCE.COM is missing or bad. (8)

ERROR: DEVICE ERROR INTERRUPT! EXITING INSTALL! - A device hardware error interrupt has occurred. (8)

ERROR: ERRNO %D. - Unknown or unexpected error condition has occurred. Contact OMATION, INC customer support with a description of conditions and ERRNO number. (9)

ERROR: FORMAT ERROR. - Manager unable to EXEC command due to an EXEC format error. (9)

ERROR: INVALID ARGUEMENT(S). - Manager unable to shell command. Invalid command line arguements encountered. (9)

ERROR: MUTUALLY EXCLUSIVE OPTIONS IN %S. - Mutually exclusive execution options or parameters have been specified for the command in the .MGR manager configuration file. (9)

ERROR: NO DRIVE SPECIFIED IN <PATH>. PLEASE ENTER COMPLETE PATHNAME! - The drive specifier was excluded from <path> or entered incorrectly. (8)

ERROR: NOT A DEDICATED COMMAND. - Command execution parameters in the manager configuration file include the "D" or DEDICATED option following the "@" but the command is not recognized by the manager. Check the command configuration, execution parameters and spelling. **(9)**

ERROR: NOT ENOUGH MEMORY. - Insufficient base memory available to shell and execute command. **(5,6,9)**

ERROR: NOT READY ERROR READING DRIVE <DRIVE>. - A disk hardware interrupt has occurred while accessing the specified drive. Check to see that a diskette is securely in the drive. **(8)**

ERROR: PATH OR FILE NAME NOT FOUND. - Manager unable to shell command. Command not found in current directory or in DOS path. Check the command spelling and the PATH parameter in the AUTOEXEC.BAT file in the root directory. **(9)**

ERROR: TOO MANY ARGUEMENT(S). - Manager unable to shell command. Too many command line arguements specified. **(9)**

ERROR: UNABLE TO ACCESS OR MAKE DIRECTORY <DIRECTORY>. - <Directory> given, or path up to <directory> to be created, does not exist. Install will not create more than one directory in a path (i.e., the last directory in the path). **(8)**

ERROR: UNABLE TO ACCESS OR WRITE CONFIGURATION FILE! - The SCHEMA.CFG configuration file is missing, not at the full path given for the install, or the DOS attribute has been changed preventing access. **(8)**

ERROR: UNABLE TO OPEN NUL STDOUT. - Unable to redirect standard output messages generated by ARCE.COM (unarchiver). **(8)**

ERROR: WRONG DISKETTE OR NO VOLUME LABEL! - The diskette in the drive from which the install is occurring has no volume label (the wrong diskette) or cannot be read. Check the diskette again before proceeding. **(8)**

EXPANDED MEMORY ERROR - SAVE AND EXIT - The expanded memory manager (EMM) detected an error in the EMM hardware. The screen may look incorrect, but the drawing is actually still intact. Save the drawing (if changes were made) and exit SCHEMA. Refer to your EMM manual for any diagnostic procedures that can be run. **(1,3)**

FAILED TO UPDATE SCHEMA.CFG - **CONFIG-SAVE** could not open the SCHEMA.CFG or the file has a problem. Check to see that the path given to find the SCHEMA.CFG was correct. **(1)**

FATAL ERRORS ENCOUNTERED - PROGRAM ABORTED - An error has been detected such that SCHPOST can no longer continue to process the drawing. This message should be preceded by another message indicating the reason for termination. **(4)**

FILE CREATE FAILED - In attempting to save a drawing to disk, a DOS open error occurred. The disk or disk directory is probably full. The save attempt was aborted. Try to save to a floppy drive. **(1)**

FILE EXISTS, PRESS <ENTER> TO OVERWRITE - The file that you are saving to already exists . Press <Enter> to confirm the overwrite. **(1)**

FILE IS MARKED READ-ONLY. SAVE ABORTED - The file name to which you are saving the drawing is read-only. Save the drawing to another name. (1)

FILE MODIFIED SINCE READ, PRESS <ENTER> TO OVERWRITE - It is possible with a network for two users to load the same schematic drawing page. If one user alters and saves the drawing before the other, then when the other user attempts to save this warning will inform the user that the drawing has been modified by another within the same drawing session. **(1)**

FILE NOT FOUND - The drawing file cannot be located on disk. It does not exist or the directory/path is incorrect. **(3)**

FILE OPEN FAILED, TOO MANY FILES OPEN - A file could not be opened because too many other files are in use. Increase the **FILES=** parameter in the **CONFIG.SYS** file. **(1,3)**

FILE OR PATH NOT FOUND - The drawing file cannot be located on disk. It does not exist or the directory/path is incorrect. **(1)**

FILE PRINT.COM NOT FOUND....PRINT ABORT - The DOS print utility, PRINT.COM, can not be found. Either the print utility does not exist or the PATH statement is incorrect. For floppy disk-based systems, the DOS PRINT.COM and COMMAND.COM programs are normally placed on the Standard Library disk. **(1)**

FILE READ ERROR - A DOS read error has occurred in reading a drawing file. The disk image is bad. **(1,2,3)**

FILE WRITE FAILED - In attempting to save a drawing to disk, a DOS write error occurred. The disk is probably full. Only part of the drawing was saved. **(1)**

GLOBAL NET OBJECT FOUND ON WRONG PAGE - A global net object was found on a page other than the first page of a top level schematic. See Chapter 7, SCHPOST. **(4)**

ILLEGAL ENTRY, PRESS ENTER - An incorrect selection was made from the Menu. **(5,6)**

ILLEGAL FILE - The given filename is not a valid SCHEMA III drawing page. **(3)**

ILLEGAL "GRAPH=" PARAMETER IN SCHEMA.CFG...ABORT - The configuration display type entry (**GRAPH=**) in the SCHEMA.CFG file is unrecognizable. Use SCHCFG to correct the **GRAPH=** variable in the SCHEMA.CFG file. **(1)**

INTERNAL ERROR - An error occured when processing the schematic data. This error should not occur under normal circumstances. **(4)**

INVALID CIRCLE PARAMETERS - The entered parameters are not valid for drawing circles/arcs. See the **BODY-LINE** command summary in Chapter 3, *SCHDRAW*. **(1)**

INVALID DRAWING FILE - The desired file is not a valid SCHEMA III drawing file. The load was aborted. **(1,3)**

INVALID DRAWING FILE - ABORT. - The drawing file specified is invalid. **(2)**

INVALID EXTENSION EXT - PAGE IGNORED. - The extension, *ext*, is not a valid drawing file extension. **(2)**

INVALID OBJECT ENTRY XX - The object definition contains the invalid entry, *xx*. The object is not processed beyond this point. **(2)**

INVALID PAGE ENTRY XX - The drawing file contains the invalid entry, *xx*. The page is not processed beyond this point. **(2)**

INVALID PRIMARY COMMAND, PRESS VALID COMMAND OR <ESC> - The selected primary command is invalid. Either press the first letter of the valid primary command or press **<Esc>**. **(1)**

INVALID REGION FILE - The desired file is not a valid SCHEMA III REGION file. The load was aborted. **(1)**

INVALID SECONDARY COMMAND, PRESS VALID COMMAND OR <ESC> - The selected secondary command is invalid. Either press the first letter of the valid secondary command or press **<Esc>**. **(1)**

NOTE
For the following messages, n = the applicable configuration file library variable (LIBR) number, 1-9.

INVALID X,Y FORMAT AND/OR DIGITS - The x,y coordinates specified in the **SEARCH** command were specified incorrectly or contain invalid digits. **(1)**

LABEL EXCEEDS 12 CHARACTERS - This error occurs when trying to create a SCHEMA PCB pin list (S option under SCHPOST) with labels greater than 12 characters. SCHEMA PCB cannot accept net names with greater than 12 characters. **(4)**

LIBRARY ACCESS DENIED - The library may be marked as read only and saving to it is denied. **(1)**

LIBRARY CANNOT BE READ - The library can not be read. The disk or file is bad. **(1,3,5)**

LIBRARY DIRECTORY FULL, SAVE ABORTED - The object can not be saved to the library because the library's directory is full. Expand the directory or use another library. **(1,5)**

LIBRARY NOT FOUND - The library assigned to the **LIBRn** variable in the associated configuration file can not be located. The library does not exist or the directory/path is incorrect. The complete path and library name (including extension) must be given. Use SCHCFG to examine the failing library variable in the configuration file. **(1,3,5)**

LIBRARY NOT OPENED, TOO MANY FILES OPEN - The maximum number of open files allowed by DOS has been exceeded. Change the FILES= entry in the CONFIG.SYS file. **(1,3,5)**

LIBRARY READ ERROR, LIBRARY INVALID - In reading a library's directory, a read error occurred. The disk or library is bad. **(1,3)**

LIBRARY WRITE ERROR, LIBRARY INVALID - An error has occurred in saving an object to a library. The disk is full or bad, or you do not have write-access rights to the stated library. **(1,5)**

MAXIMUM INCREMENT SIZE IS 32 - The maximum vertical TEXT increment value is 32. Reenter text with a smaller increment value. **(1)**

MEMORY LIMITS REACHED, TRY REDRAW OR SAVE AND EXIT - The size of the drawing page and the number of objects have exceeded memory limits. No additional items can be added to the drawing. Continue the drawing on another page. If you are using a 512K PC and you receive this message frequently, consider upgrading your memory to 640K. With 640K, under normal operation, this message is rarely displayed. In addition, if you are using memory-resident programs, i.e., print spoolers, keyboard enhancers, etc. you may need to remove them.

If the problem occurs on a 640K PC, consider using expanded memory. (1)

MONOCHROME CONFIGURED ADAPTER. . .COLOR NOT SUPPORTED - The ZOOM-VGA switch only works for those with higher resolution graphics adapters with VGA support. (1)

<NAME> OBJECT NOT FOUND - The object <NAME> can not be found. The object does not exist in any library. (1,3,5)

NAME OBJECT AND BODY OBJECT HAVE THE SAME NAME - It is illegal for a name object to reference a body object of the same name. See Chapters 1 and 3 for the proper procedure to use when creating a part. (4)

NET NAME NOT FOUND IN GLOBAL NET OBJECT - A net name was not specified as the first parameter of the GLOBAL object. See Chapter 7, SCHPOST. (4)

NETS WITH A SINGLE PIN - This error occurs when trying to create a SCHEMA PCB pin list (S option under SCHPOST) with single pin nets. The SCHECO program cannot function properly with single pin nets. (4)

NEW COLORS/PAN, PRESS <ENTER> TO UPDATE SCHEMA.CFG - When CONFIG-SAVE is selected from SCHDRAW, if the colors or the panning speed are altered, these are considered hardware changes and are saved to the SCHEMA.CFG. (1)

NO "GRAPH=" IN SCHEMA.CFG...ABORT - The graphics adapter configuration file entry (GRAPH=) is missing from the SCHEMA.CFG file. This entry must be present. Use SCHCFG to add a GRAPH= entry to the SCHEMA.CFG file. (1)

NO FILES FOUND - No files were found that compared with the search string. **(1)**

NO "LIBR1=" IN CONFIGURATION FILE...ABORT - No entry for library one is in the associated configuration file. At a minimum, the first library, **LIBR1**, must be assigned. Use SCHCFG to assign an object library to the configuration file **LIBR1** variable. **(1,3)**

NO MORE MATCHES - No more items can be found on the page that match the search string specified for the **SEARCH** command. **(1)**

NO OBJECT FOUND - No objects were found that compared with the search string. **(1)**

NO OTHER GATES/SECTIONS FOUND - The **GATE** command of SCHDRAW cannot find the next consecutive letter of the next gate in any of the libraries. **(1)**

NO OTHER ROTATION FOUND - The **ROTATE** command of SCHDRAW cannot find the next consecutive number of the next rotation in any of the libraries. **(1)**

NO PAGE FILE SPECIFIED - A schematic page was not specified when executing SCHPOST. **(4)**

NO "PRINT=" IN SCHEMA.CFG...ABORT - The printer configuration entry (**PRINT=**) is absent from the SCHEMA.CFG configuration file. *Even if a printer is not to be used, a print driver must be identified.* Use SCHCFG to add a **PRINT=** variable to the SCHEMA.CFG file. **(1,3)**

NOT ENOUGH MEMORY, ABORTED - Not enough memory to run SCHPOST. See Appendix E. **(4)**

NOT ENOUGH MEMORY...ABORT - There is not enough memory available to load a default drawing page. If you are using a 512K PC and you receive this message frequently, consider upgrading your memory to 640K. With 640K, this message is seldom displayed with SCHEMA III supplied pages. However, if you are using memory-resident programs, i.e., print spoolers, keyboard enhancers, etc. you may need to remove them. Also, large **BUFFERS=** and **FILES=** statements in the DOS CONFIG.SYS file can use excessive memory. Use the DOS CHKDSK program to identify the amount of free memory in your PC. With 640K systems, SCHEMA III requires at least 590K of free memory. If the problem occurs on a 640K PC, consider using expanded memory. **(1,3)**

NOT FOUND IN TABLE: <REFD> - The reference designator **<REFD>** could not be located in the SCHBACK 'was/is' table (**FILENAME.REF** file). The reference designator is probably missing or it has an incorrect sub-part (gate) section identifier. The **<REFD>** reference designator was not updated. **(7)**

OBJECT ALREADY PRESENT, PRESS <ENTER> TO OVERWRITE - The object being saved to disk already exists. To overwrite this object, press **<Enter>**. Else press any other key. **(1)**

OBJECT CANNOT HAVE THE SAME NAME AS NESTED OBJ - You are attempting to save a Body Object under the same name as a Nested Object within the current Body Object, i.e., the object is calling itself. Assign a unique name to this Body Object. **(1)**

OBJECT NAME TOO LONG, MAXIMUM OF 8 CHARS - The length of the object name exceeds eight characters. Shorten the object name and rename the object. **(1)**

OBJECT 'OBJ' NOT FOUND - Object, *obj*, cannot be found in any of the libraries specified in the configuration file. **(2)**

OBJECT/REGION TOO LARGE - GOTO HIGHER ZOOM LEVEL - The object or region is too large to fit in the object buffer. Use a higher zoom level to work with the object or region. If this message is displayed during a **MOVE**, issue **UNDO** to restore the image. **(1)**

OBJECT TOO LARGE FOR PAGE SIZE - The object is larger (in one or both directions) than the current page. The object can not be placed on the current page. **(1)**

OUT OF RANGE, OFF PAGE - The **REPEAT** operation can not be used because the image would be placed outside the current drawing page. **(1)**

PAGE DRAWING.PXX NOT FOUND - PAGE IGNORED. - Page, *drawing.Pxx*, cannot be found. Either the name or the path is invalid or too many files are open. The number of files allowed to be open can be increased by changing the **FILES=** statement in your CONFIG.SYS file. **(2)**

PAGE DRAWING.P01 NOT FOUND - PAGE IGNORED. - Page, *drawing.P01*, cannot be found. Either the name or the path is invalid or too many files are open. The number of files allowed to be open can be increased by changing the **FILES=** statement in your CONFIG.SYS file. **(2)**

PAGE FAULT DATA PROBABLY LOST - An error has been detected that probably corrupted the current drawing. Try saving the drawing under a temporary name and exit SCHDRAW. Then, reenter SCHDRAW and determine if the saved file is usable. This error should not occur under normal operation. **(1)**

PAGE TOO LARGE, NOT ENOUGH MEMORY - Not enough memory is present to load the drawing. If you are using a 512K PC and you receive this message frequently, consider upgrading your memory to 640K. With 640K, this message is seldom displayed with SCHEMA III supplied pages. However, if you are using memory-resident programs, i.e., print spoolers, keyboard enhancers, etc. you may need to remove them. Also, large **BUFFERS=** and **FILES=** statements in the DOS CONFIG.SYS file can use excessive memory. Use the DOS CHKDSK program to identify the amount of free memory in your PC. With 640K systems, SCHEMA III requires at least 590K of free memory. If the problem occurs on a 640K PC, consider using expanded memory. **(1,2,3)**

PARAMETER DOES NOT EXIST FOR PART - The parameter used in a pin definition of the part does not exist. **(4)**

PIN NAME AND NET MUST BE SPECIFIED - The BODY-PIN commnand when used with the '@' icon must have the pin and the net name assigned before a hidden power or ground pin can be created. **(1)**

PLOT ABORTED BY OPERATOR - The operator aborted the plot output by pressing the **<Esc>** key. **(2)**

PRINT HANDLER NOT FOUND...ABORT - The print driver defined in the SCHEMA.CFG configuration file **PRINT=** variable can not be located on disk. It does not exist or the directory/path is incorrect. The **PRINT=** variable must be assigned a complete path name and file name (including extension) even if no printer is used. Use SCHCFG to redefine the **PRINT=** variable in the SCHEMA.CFG file. **(1,3)**

PRINT I/O - DISK FULL...ABORT - In performing a print operation, DOS returned a write error. This typically implies that the disk being written to is full. The print operation is aborted. **(1,3)**

PRINT I/O - WRITE ERROR...ABORT - In performing a print operation DOS returned an I/O error. This typically implies some sort of hardware error associated with the disk being written. **(1,3)**

PRINT SPOOL ERROR...SPOOLER NOT ACTIVE - An error occurred when attempting to execute the print spooler (DOS PRINT program). The spool may or may not have been carried out. The DOS PRINT command must be invoked prior to executing SCHDRAW. **(1)**

PRINT SPOOL ERROR...UNABLE TO QUEUE FILE - An error occurred when attempting to queue the file to the print spooler (DOS PRINT program). The spool may or may not have been carried out. The DOS PRINT command must be invoked prior to executing SCHDRAW. **(1)**

PROCESSING PAGE: <FILENAME.PXX> - This is a user message (not an error). The page **FILENAME.PXX** is currently being processed by SCHBACK. **(7)**

REGION EXISTS, PRESS <ENTER> TO OVERWRITE - The desired **REGION** file already exists. Press **<Enter>** to overwrite the file. **(1)**

REGION IS EMPTY - The region defined contains no items to perform any region operations . **(1)**

REGION LARGER THAN PAGE, CANNOT LOAD - The desired **REGION** file is larger than the current drawing page. The load was aborted. **(1)**

REGION TOO LARGE, GOTO HIGHER ZOOM LEVEL - The region is too large to use in this perspective. If already in the 8:1 **ZOOM** mode, the region is too large and must be divided into two or more sections. If you are using a 512K PC and you receive this message frequently,

consider upgrading your memory to 640K. With 640K, under normal operation, this message is rarely displayed. In addition, if you are using memory-resident programs, i.e., print spoolers, keyboard enhancers, etc. you may need to remove them. (1)

SELECTED PLOTTER DOESN'T SUPPORT PARTS OUTPUT - OPTION IGNORED - The /P option was specified for a plotter driver that does not support it. The option is ignored. (2)

SELECTED PLOTTER DOESN'T SUPPORT ROTATE - OPTION IGNORED - The /R option was specified for a plotter driver that does not support it. The option is ignored. (2)

SIGNAL NET NOT ATTACHED TO PIN ON PART - This message indicates that the named hidden signal net (i.e. GND) is not attached to any pin. This would generally warn the user that the named part has created incorrectly. (4)

SIGNAL NET NOT DEFINED AS GLOBAL - One of the nets, +5 or GND, was found as an internal net in a hierarchical page without being declared global. See Chapter 7, SCHPOST. (4)

SIGNAL NOT FOUND IN HIERARCHICAL DRAWING - A label matching the pin definition of a hierarchical object is not found in the corresponding hierarchical page. See Chapter 7 for a detailed explanation on the implementation of hierarchy. (4)

STACK FULL: UNABLE TO PUSH HIERARCHICAL FILE NAME - The maximum level of hierarchical files that can be held in the stack has been exceeded. (1)

TABLE ENTRY NOT USED: <WAS,IS> - The <WAS,IS> pair from the FILENAME.REF file was not used in any replacement. This is

probably an error unless you intentionally have extra reference designator pairs in your **FILENAME.REF** file. **(7)**

TOO MANY CONNECTIONS FOR ONE NET - Too many connections to a single net exist on a single page. Try splitting the net up over several pages. This error message should not occur under normal circumstances. **(4)**

TOO MANY FILES OPEN, SAVE ABORTED - The object can not be saved to the library because a new write version of the library can not be opened. The maximum number of open files allowed by DOS has been exceeded. Change the FILES= entry in the CONFIG.SYS file. **(1)**

UNABLE TO CREATE/WRITE BACKUP.P01 IN OMATION PATH - The Auto Backup feature is trying to save the backup to the file BACKUP.P01 in the path specified by the OMATION environment variable. Check to make sure the path is correct and the directory has write privaledges. **(1)**

UNABLE TO FIND DRAWING.CFG OR SCHEMA.CFG - ABORT. - The *drawing.CFG* or SCHEMA.CFG files could not be found. **(2)**

UNABLE TO LOCATE HIERARCHICAL OBJECT IN ANY OF THE SPECIFIED LIBRARIES - A hierarchical object used in a drawing cannot be found in any of the libraries specified in the configuration file. **(4)**

UNABLE TO LOCATE OBJECT IN ANY OF THE SPECIFIED LIBRARIES - An object used in a drawing cannot be found in any of the libraries specified in the configuration file. **(4)**

UNABLE TO OPEN FILE - The files cannot be opened. Make sure that the path is correct and check the **FILES=** statement in CONFIG.SYS. **(4)**

UNABLE TO OPEN HIERARCHICAL FILE - The hierarchical page file cannot be opened. Make sure that the hierarchical page is in the same directory as the library containing the corresponding hierarchcial object. Also check the **FILES=** statement in CONFIG.SYS. **(4,7)**

UNABLE TO OPEN LIBRARY - A library listed in the configuration file cannot be opened. Make sure that the path is correct and check the **FILES=** statement in CONFIG.SYS. **(4)**

UNABLE TO OPEN NET LIST FILE - SCHPOST was unable to create a new file for the netlist. Make sure that there is sufficient room on the hard disk and check the **FILES=** statement in the CONFIG.SYS. **(4)**

UNABLE TO OPEN PIN LIST FILE - SCHPOST was unable to create a new file for the pin list. Make sure that there is sufficient room on the hard disk and check **FILES=** statement in CONFIG.SYS. **(4)**

WARNING: CURRENT FILE NOT SAVED - The schematic page has been altered since it was loaded. **(1)**

WARNING. PRINTER DRIVER NOT FOUND AT SPECIFIED LOCATION. PRESS ENTER - The printer driver path is not the same as the path that was specified during installation or specified driver was never installed on PC. **(6)**

WARNING: NOT ENOUGH MEMORY TO RUBBERBAND ALL WIRES - The dynamic rubberbanding cannot display all wires due to lack of memory. However when the region is placed all the wires will be displayed. **(1)**

WRITE ERROR, SAVE ABORTED - The drawing file that is being saved, is restricted or has errors being written. **(1)**

Appendix E. Hardware Requirements

E.1 INTRODUCTION

This appendix identifies the hardware required to run SCHEMA III. It also discusses individual differences in using various graphics adapters, printers, plotters and expanded memory.

E.2 PC

SCHEMA III runs on any IBM and most IBM-compatible personal computers including the IBM PC/XT/AT/Portable PC, COMPAQ PORTABLE/DESKPRO/386, AT&T 6300, and many others.

The minimum amount of memory required is 512K. However, 640K (590K *free* RAM or more) is strongly recommended with 4M of Hard Disk space. (The PPAGE3.P99 and BPAGE3.P99 drawing page files require 640K.). Systems with 512K of memory may experience region and page size limitations directly associated with memory. If you have a 512K system and receive the REGION TOO LARGE, GOTO HIGHER ZOOM LEVEL or MEMORY LIMITS REACHED, SAVE AND EXIT messages often, consider adding additional memory to eliminate this problem. Free RAM can also be increased by reducing the **BUFFERS=** statement in the PC's **CONFIG.SYS** file. The minimum that buffers can equal is 2. Each buffer uses 520 bytes. SCHDRAW automatically looks for any expanded memory configured in your system. It compares the amount of *free* RAM to expanded memory and chooses the greater of the two. All expanded memory boards used must follow

LIM standard version 3.2 and above (LOTUS-INTEL-MICROSOFT). If you have other memory-resident programs, i.e., print spoolers, keyboard enhancers, desktops, etc. you may require 640K. You may not be able to run SCHEMA III with large print spoolers or other large memory-resident programs installed.

E.3 GRAPHICS ADAPTER

SCHEMA III supports several graphics adapters. The SCHEMA.CFG file **GRAPH=** variable specifies which graphics driver your systems requires. Actually, only SCHDRAW requires this information - all other programs work through standard DOS display drivers. There are no driver files associated with graphics adapters.

Depending on the graphics adapter you use, a varying amount of resolution is available to SCHDRAW. Thus, you will see a larger or smaller portion of a given drawing depending on the graphics adapter used. Also, since each graphics adapter has a different aspect ratio (number of X-pixels divided by number of Y-pixels), a varying amount of drawing elongation may be noticed. Regardless of the graphics adapter used, the printed (or plotted) drawing is always drawn accurately, faithful to a square aspect ratio.

The following graphics drivers are provided for the mentioned graphics adapter or its compatible. Names referenced within parentheses are the names used in the SCHEMA.CFG file to specify graphics drivers. Any newly-supported graphics drivers are listed in the READ.ME file on the SCHEMA III Program disk.

Graphics Adapter	Driver
AT & T 6300 High-Res Graphics	AT
ATI Wonder Card 800 x 600	A1
ATI Wonder Card 1024 x 768	A2
Genoa Super EGA 640 x 480	G1
Genoa Super EGA Hi-Res 800 x 600	G2
Genoa Super VGA 1024 x 768	G3
Hercules Monochrome Graphics	HE
HP Vectra High-Res Graphics	VE
Trident - Impact III Series 800 X 600	I1
Trident - Impact III Series 1024 x 768	I2
IBM Standard Color Graphics	CG
IBM Enhanced Color Graphics	EG
Multi-Color Graphics Array	MC
NEC MultiSync	NE
Orchid Pro-Designer 800 x 600/SOTA	O1
Orchid Pro Designer 1024x768/ SOTA	O2
Paradise AutoSwitch EGA 480	P1
Paradise VGA Plus 800 x 600	P2
Paradise VGA 1024 x 768	P3
Sigma Designs Color 400	SI
STB VGA 16 800 x 600	S1
STB VGA 16 1024 x 768	S2
Tecmar Graphics Master	T1
Tecmar VGA/AD 800 x 600	T2
Tecmar VGA/AD 1024 x 768	T3
Toshiba T3100 High-Res Graphics	31
Trident Impact III 800 x 600	I1
Trident Impact III 1024 x 768	I2

Video Graphics Array	VG
Video 7 (720 x 540)	V1
Video 7 (800 x 600)	V2
Video 7 VRAM 1024 x 768	V3
Wyse WY-700	WY

AT&T 6300 High-Resolution Graphics Mode (AT) - This driver provides 640 by 400 pixel monochrome graphics for AT&T Model 6300 PCs. The AT&T 6300 can also emulate the Standard IBM Color Graphics Adapter (CGA).

ATI Wonder Card 800 x 600 (A1) - This driver provides eight-color 800 x 600 graphics. Refer to EGA paragraph for more information on color configuration.

ATI Wonder Card 1024 x 768 (A2) - This driver provides eight-color 1024 x 768 graphics. Refer to EGA paragraph for more information on color configuration.

Genoa 640 x 480 (G1) - This driver provides eight-color 640 by 480 graphics. Refer to the EGA paragraph for information on color configuration.

Genoa 800 x 600 (G2) - This driver provides eight-color 800 by 600 graphics. Refer to the EGA paragraph for information on color configuration.

Genoa 1024 x 768 (G3) - This driver provides eight-color 1024 by 768 graphics. Refer to the EGA paragraph for information on color configuration.

Hercules Monochrome Graphics Adapter (HE) - This driver provides 720

by 348 pixel monochrome graphics. Other manufacturers offer Hercules-compatible adapters which may work with SCHEMA III. SCHEMA III only uses the first page (32K) of memory on this adapter.

HP Vectra High-Resolution Graphics Mode (VE) - This driver provides 640 by 400 pixel monochrome graphics for HP Vectra PCs. The HP Vectra can also emulate the Standard IBM Color Graphics Adapter (CG).

Trident Impact III 800 x 600 (I1) - This driver provides eight-color 800 by 600 graphics. Refer to the EGA paragraph for information on color configuration.

Trident Impact III 1024 x 768 (I2) - This driver provides eight-color 1024 by 768 graphics. Refer to the EGA paragraph for information on color configuration.

IBM Standard Color Graphics Adapter (CG) - This driver is for the original IBM graphics adapter. It is used in the 640 by 200 pixel black and white mode. Some computers such as the IBM Portable PC, COMPAQ PORTABLE and DESKPRO, Zenith, etc. have this adapter built-in. If you are not sure which graphics adapter you have, try this one first.

IBM Enhanced Graphics Adapter (EG, 64KEGA) - This driver supports the IBM Enhanced Graphics Adapter (EGA). EG supports 128K/256K EGA cards (most EGA cards). 64KEGA supports the 64K EGA card (early IBM). All EGA cards are used in the 640 by 350 pixel mode. Switch settings on the adapter determine if it drives a color (operating at 22 kHz) or monochrome (operating at 18 kHz) display. If you are using a standard (15 kHz) black and white or color monitor with this adapter, it is operating in the IBM Standard Color Graphics mode and CG is the

graphics adapter type to specify in the SCHEMA.CFG file. If you specify the EG entry, you can change the eight screen colors by following the SCHCFG EG entry with a comma (,) and palette number entry. Each palette number entry is a hex number from 0 to 3F. The entry bits are:

R, G, B, R', G', and B'. Refer to the SCHCFG GRAPH= command description in Chapter 5 for additional information. The 64KEGA entry runs the 64K EGA card in monochrome.

Multi-Color Graphics Array (MCGA) - this driver is supported only in 640 by 480 pixel monochrome mode.

NEC MultiSync (NECEGA) - This driver provides eight-color 640 by 480 graphics. Refer to the EGA paragraph for information on color configuration.

Orchid Pro-Designer 800 x 600 / SOTA (O1) - This driver supports 800 x 600 mode.

Orchid Pro-Designer 1024 x 768 / SOTA (O2) - This driver supports 800 x 600 mode.

Paradise AutoSwitch EGA 480 (P1) - This driver provides eight-color 640 by 480 graphics. Refer to the EGA paragraph for information on color configuration.

ParadiseVGA Plus 800 x 600 (P2) - This driver provides eight-color 800 by 600 graphics. Refer to the EGA paragraph for information on color configuration.

Paradise VGA Plus 1024 x 768 (P3) - This driver provides eight-color 1024 by 768 graphics. Refer to the EGA paragraph for information on color configuration.

Sigma Designs Color 400 (SI) - This driver is used in the 640 by 400 pixel monochrome mode. The Color 400 can also emulate the Standard IBM Color Graphics Adapter (CG). You can change the screen color by following the SCHCFG SIGMA entry with a comma (,) and palette number (single hex digit from 0 to F). Refer to your Color 400 owner's manual for a listing of available screen colors and the applicable hex digit entry.

STB VGA 16 (S1) - This driver provides eight-color 800 x 600 graphics. Refer to EGA.

STB VGA 16 + (S2) - This driver provides eight-color 1024 x 768 graphics. Refer to EGA.

Tecmar Graphics Master (T1) - This driver is used in the 640 by 400 pixel monochrome mode (Setup #1). The Graphics Master can also emulate the Standard IBM Color Graphics Adapter (CG). You can change the screen color by following the SCHCFG TECMAR entry with a comma (,) and palette number (single hex digit from 0 to F). Refer to your Graphics Master owner's manual for a listing of available screen colors and the applicable hex digit entry.

Tecmar VGA 800 x 600 (T2) - This driver provides eight-color 800 by 600 graphics. Refer to the EGA paragraph for information on color configuration.

Tecmar VGA 1024 x 768 (T3) - This driver provides eight-color 1024 by 768 graphics. Refer to the EGA paragraph for information on color configuration.

Toshiba T3100 High-Resolution Graphics Mode (31) - This driver

provides 640 by 400 pixel monochrome graphics for the Toshiba T3100 PC.

Trident Impact III 800 x 600 (I1) - This driver provides eight-color 800 by 600 graphics. Refer to the EGA paragraph for information on color configuration.

Trident Impact III 1024 x 768 (I2) - This driver provides eight-color 1024 by 768 graphics. Refer to the EGA paragraph for information on color configuration.

Video Graphics Array (VG) - This driver provides eight-color 640 by 480 graphics. Refer to the EGA paragraph for information on color configuration.

Video 7 (V1) - This driver provides eight-color 720 by 540 graphics. Refer to EGA.

Video 7 (V2) - This driver provides eight-color 800 by 600 graphics. Refer to EGA.

Video 7 (V3) - This driver provides eight-color 1024 by 768 graphics. Refer to EGA.

Wyse WY-700 (WY) - This driver provides 1280 by 800 pixel monochrome resolution for the Wyse WY-700.

E.4 MOUSE

SCHDRAW supports any mouse that has a Microsoft-compatible mouse driver. This software program or driver should have been supplied with your mouse. Almost all mouse manufacturers including Microsoft, Mouse Systems, Logitech, Summagraphics, and Torrington support this driver. Refer to your mouse manual for information on installing the Microsoft-compatible mouse driver. If your mouse did not come with a driver, contact the manufacturer.

Mouse sensitivity (speed) for SCHDRAW can be adjusted using the SCHCFG program. Refer to Chapter 5, *SCHCFG*, for additional information. The sensitivity for the Manager can be set by selecting the EDIT command from the Control section.

There is no manufacturer-unique information relating to SCHDRAW mouse support. SCHDRAW uses the left mouse button to select commands, while the right mouse button is used exclusively to activate the SCHDRAW **UNDO** command. If you have a three-button mouse, the middle button is used to freeze the Drawing Window cursor's position and activate the Command Window Indicator to select commands - refer to Chapter 3 for additional information on SCHDRAW mouse operation.

E.5 PRINTER

SCHEMA III supports various printer types. The printer type is defined in the SCHEMA.CFG file with the **PRINT=** variable. This variable is only used by SCHDRAW and SCHPRINT, i.e., for printing drawings. The other SCHEMA III programs that produce output do not use this variable and thus, use only standard print formatting (line feeds, form feeds, and carriage returns). There is a printer device driver file for each printer type. This file is generally located in the same directory as the SCHEMA III program files. See Appendix A, *Installing SCHEMA III Manually*, for specific information on file configuration.

PostScript devices are supported only through SCHPLOT.

Printing can be redirected to any COM or LPT port.

The next page contains a list of available printers and their corresponding printer drivers. As well as the following pages discuss each printer supported by SCHEMA III. Names referenced in parentheses are the driver names used in the SCHEMA.CFG file to specify printer type. Any newly-supported printer types are listed in the READ.ME file on the SCHEMA III Program disk.

Printer	Driver
Apple Imagewriter Wide	IMAGE15.DRV
Apple Imagewriter Narrow	IMAGE8.DRV
C-Itoh 1550S	CI1550.DRV
C-Itoh 8510S	CI8510.DRV
Canon LBP-8 A2	ICANON.DRV
Citizen MSP 10	CITMSP10.DRV
Citizen MSP 15	CITMSP15.DRV
Epson Wide-Carriage	EPSON100.DRV
Epson Narrow-Carriage	EPSON80.DRV
Epson LQ-Series Wide	ILQ1500.DRV
Epson LQ-Series Narrow	LQ2500.DRV
Genicom 3000 Series	GENICOM.DRV
Hewlett-Packard Deskjet	DESKJET.DRV
Hewlett-Packard LaserJet+	IHPLAS.DRV
Hewlett-Packard PaintJet	HPTJET1.DRV
IBM Color Printer	IBMCOLOR.DRV
Null (raster output)	NULL.DRV
Okidata 82A	OKI82A.DRV
Okidata 83A	OKI83A.DRV
Okidata 193 Wide-Carriage	IOKI193.DRV
Okidata 192 Narrow-Carriage	IOKI192.DRV
Okidata 2410 Wide-Carriage	OKI2410W.DRV
Okidata 2410 Narrow-Carriage	OKI2410N.DRV
Okidata ML92	OKIML92.DRV
Okidata ML93	OKIML93.DRV

Printer	Driver
PC Paintbrush	PCX.DRV
Printronix 300/600	PTRON600.DRV
Proprinter Wide-Carriage	PRO100.DRV
Proprinter Narrow-Carriage	PRO80.DRV
Star Wide-Carriage	STAR100.DRV
Star Narrow-Carriage	STAR80.DRV
Texas Instruments	TI850.DRV
ThinkJet (HP Mode)	HPTJET1.DRV
ThinkJet (Non-HP Mode)	HPTJET2.DRV
TIFF	TIFF.DRV
Toshiba Wide-Carriage	ITOS351.DRV
Toshiba Narrow-Carriage	ITOS340.DRV

PostScript Printers use SCHPLOT

Apple Imagewriter Wide-Carriage (**IMAGE15.DRV**) - This driver supports the Apple Imagewriter 15-inch wide carriage printer. The printer is configured for the following print modes and dot resolutions (horizontal by vertical):

A Mode - 120 x 96 dpi
C Mode - 160 x 144 dpi
D Mode - 60 x 48 dpi

Apple Imagewriter Narrow-Carriage (**IMAGE8.DRV**) - This driver supports the Apple Imagewriter 8-inch narrow-carriage printer.

This driver works identically to the Apple Imagewriter wide-carriage driver except some printouts are divided into two or more pages because of smaller platen size.

C-Itoh 1550S (**CI1550.DRV**) - This driver supports the C-Itoh 1550S 15.5-inch wide-carriage dot matrix printer.

Mode A - 136 x 144 dpi
Mode C - 80 x 80 dpi (low resolution)
Mode D - 68 x 72 dpi (high resolution)

C-Itoh 8510S (**CI8510.DRV**) - This driver supports the C-Itoh 8510S 10-inch narrow dot matrix printer.

Canon LBP-8 A2 (**ICANON.DRV**) - This driver supports the Canon LBP-8 A2 laser printer (you must have the *A2* model).

The printer is configured for the following print modes and dot resolutions (horizontal by vertical):

A Mode - 150 x 150 dpi (letter size paper)
C Mode - 300 x 300 dpi (letter sized paper)
D Mode - 150 x 150 dpi (legal size paper)

Citizen MSP 15 (**CITMSP15.DRV**) - This driver supports the Citizen MSP 15-inch wide-carriage printer.

A Mode - 120 x 108 dpi
C Mode - 144 x 144 dpi
D Mode - 72 x 72 dpi

Citizen MSP 10 (**CITMSP10.DRV**) - This driver supports the Citizen MSP 10-inch narrow-carriage printer.

A Mode - 120 x 108 dpi
C Mode - 144 x 144 dpi
D Mode - 72 x 72 dpi

Epson Wide-Carriage (**EPSON100.DRV**) - This driver supports the Epson 100-series of 15-inch wide-carriage printers including MX100 (with GRAFTRAXPLUS), RX100, FX100, FX100+, FX185, FX286, etc. There are many printers offered by other manufacturers that are compatible with Epson printers. These manufacturers include IBM, Citizen, Canon, Panasonic, etc. The printer is configured for the following print modes and dot resolutions (horizontal by vertical):

A Mode - 120 x 108 dpi
C Mode - 60 x 72 dpi (low resolution, high speed)
D Mode - 60 x 54 dpi (high resolution)

The check mode available in SCHEMA II has been eliminated. To achieve the same results, use the "C Mode" along with the 2:1 zoom level.

Epson Narrow-Carriage (**EPSON80.DRV**) - This driver supports the Epson series of 10-inch narrow-carriage printers including MX80 (with GRAFTRAXPLUS), RX80, FX80, FX80+, and FX85. The other manufacturers mentioned under Epson Wide-Carriage also offer similar 10-inch printers.

This driver works identically to the Epson wide-carriage driver except printouts are often divided into two or more pages because of smaller platen size.

Epson LQ-Series Wide-Carriage (**ILQ1500.DRV**) - This driver supports the Epson LQ-series printers including the LQ1500 and LQ1000 wide-carriage printers. The printer is configured for the following print modes and dot resolutions (horizontal by vertical):

A Mode - 120 x 120 dpi
C Mode - 180 x 180 dpi
D Mode - 60 x 60 dpi

Epson LQ-Series Narrow-Carriage (**LQ2500.DRV**) - This driver supports the Epson LQ-series narrow-carriage printers including the 2500. The printer is configured for the following print modes and dot resolutions (horizontal by vertical):

A Mode - 120 x 120 dpi
C Mode - 180 x 180 dpi
D Mode - 60 x 60 dpi

Genicom 3000 Series (GENICOM.DRV) - This driver supports the GENICOM 3000 series printers. The printer is configured for the following print modes and dot resolutions (horizontal by vertical):

A Mode - 144 x 144 dpi
C Mode - 72 x 72 dpi (low resolution)
D Mode - 72 x 72 dpi (high resolution)

Hewlett-Packard Deskjet (DESKJET.DRV) - This driver supports the Hewlett-Packard (HP) Deskjet. The printer is configured for the following print modes and dot resolutions (horizontal by vertical):

A Mode - 150 x150 dpi (letter size paper)
B Mode - 300 x 300 dpi (letter size paper)
C Mode - 300 x 300 dpi (legal size paper)
D Mode - 150 x 150 dpi (legal size paper)
E Mode - 75 x 75 dpi (letter size paper)
F Mode - 75 x 75 dpi (legal size paper)
G Mode - 100 x 100 dpi (letter size paper)
H Mode - 100 x 100 dpi (legal size paper)

Hewlett-Packard LaserJet+ (IHPLAS.DRV) - This driver supports the Hewlett-Packard (HP) LaserJet+ laser printer (you must have the *PLUS* model). The printer is configured for the following print modes and dot resolutions (horizontal by vertical):

A Mode - 150 x150 dpi (letter size paper)
B Mode - 300 x 300 dpi (letter size paper)
C Mode - 300 x 300 dpi (legal size paper)
D Mode - 150 x 150 dpi (legal size paper)
E Mode - 75 x 75 dpi (letter size paper)
F Mode - 75 x 75 dpi (legal size paper)
G Mode - 100 x 100 dpi (letter size paper)
H Mode - 100 x 100 dpi (legal size paper)

Hewlett-Packard PaintJet (HPPJET.DRV) - This printer driver supports the Hewlett-Packard PaintJet printer. Refer to the Hewlett-Packard LaserJet +. The printer is configured for two modes:

Mode A - 180 x 180 dpi
Mode D - 90 x 90 dpi

Hewlett-Packard ThinkJet (HPTJET1.DRV) - This printer driver supports the Hewlett-Packard ThinkJet printer set to HP mode. The printer is configured for one mode:

Mode A - 96 x 96 dpi

Hewlett-Packard ThinkJet (**HPTJET2.DRV**) - This printer driver supports the Hewlett-Packard ThinkJet printer set to Non-HP mode. The printer is configured for one mode:

Mode A - 96 x 96 dpi

IBM Color Printer (**IBMCOLOR.DRV**) - This printer driver supports the IBM Color Printer. The printer is configured for the following print modes (horizontal by vertical):

A Mode - 168 x 168 dpi
C Mode - 84 x 84 dpi (low resolution)
D Mode - 84 x 84 dpi (high resolution)

Okidata 82A (**OKI82A.DRV**) - This driver supports the Okidata 82A printer.

A Mode - 60 x 66 dpi

Okidata 83A (**OKI83A.DRV**) - This driver supports the Okidata 83A printer.

A Mode - 60 x 66 dpi

Okidata 193 Wide-Carriage (**IOKI193.DRV**) - This driver supports the standard (non-IBM) Okidata 193 15-inch wide-carriage printer. If you own an Okidata 193 IBM-compatible printer, use the EPSON100.DRV driver. The Okidata printer is usually configured for **7** data bits. SCHEMA III uses an **8** data bit configuration. Make sure to change your printer accordingly. The printer is configured for the following print modes and dot resolutions (horizontal by vertical):

A Mode - 120 x 108 dpi
C Mode - 144 x 144 dpi
D Mode - 72 x 72 dpi

Okidata 192 Narrow-Carriage (**IOKI192.DRV**) - This driver supports the standard (non-IBM) Okidata 192 10-inch narrow-carriage printer. If you own an Okidata 192 IBM-compatible printer use the EPSON80.DRV driver. The Okidata printer is usually configured to accept 7 data bits. SCHEMA III uses an 8 data bits configuration. Make sure to change your printer accordingly.

This driver works identically to the Okidata 193 wide-carriage driver except some printouts are divided into two or more pages because of smaller platen size.

Okidata 2410 Wide-Carriage (**OKI2410W.DRV**) - This driver supports the Okidata 2410 wide-carriage printer. The printer is configured for the following print modes (horizontal by vertical):

A Mode - 120 x 108 dpi
C Mode - 144 x 144 dpi
D Mode - 72 x 72 dpi

Okidata 2410 Narrow-Carriage (**OKI2410N.DRV**) - This driver supports the Okidata 2410 narrow-carriage printer. The printer is configured for the following print modes:

A Mode - 120 x 108 dpi
C Mode - 144 x 144 dpi
D Mode - 72 x 72 dpi

Okidata ML92 (**OKIML92.DRV**) - This driver supports the Okidata ML92 narrow-carriage printer.

A Mode - 72 x 72 dpi

Okidata ML93 (**OKIML93.DRV**) - This driver supports the Okidata ML93 wide-carriage printer.

A Mode - 72 x 72 dpi

PC Paintbrush (**PCX.DRV**) - This driver supports the PC Paintbrush PCX format. Like the TIFF driver the user must redirect the output to a file using the **/O** option in **SCHPRINT**. The modes for PCX are the same as the modes described for TIFF.

Proprinter Wide-Carriage (**PRO100.DRV**) - This driver supports the Proprinter. The printer is configured the same as the Epson Wide-Carriage. Refer to *Epson Wide-Carriage* for futher information.

Proprinter Narrow-Carriage (**PRO80.DRV**) - This driver supports the Proprinter. The printer is configured the same as the Epson Narrow-Carriage. Refer to *Epson Narrow-Carriage* for further information.

Printronix 300/600 (**PTRON600.DRV**) - This driver supports the Printronix printer. The printer is configured for the following print mode and dot resolution (horizontal by vertical):

A Mode - 60 x 72

Star Wide-Carriage (**STAR100.DRV**) - This driver supports the Star 100-series of 15-inch wide-carriage printers and others similar to Epson printers. The printer is configured for the following print modes and dot resolutions (horizontal by vertical):

A Mode - 120 x 108 dpi
C Mode - 60 x 72 dpi (low-resolution, high speed)
D Mode - 60 x 48 dpi (high-resolution)

The check mode available in SCHEMA II has been eliminated. To achieve the same results, use the "C Mode" along with the 2:1 zoom level.

Star Narrow-Carriage (**STAR80.DRV**) - This driver supports the Star series of 10-inch narrow-carriage printers. This driver works identically to the Star wide-carriage driver except printouts are often divided into two or more pages because of smaller platen size.

Texas Instruments 850 (TI850.DRV) - This driver supports the Texas Instruments 850 printer. The printer is configured for the following print modes and dot resolutions (horizontal by vertical):

A Mode - 120 x 108 dpi
C Mode - 60 x 72 dpi (low resolution, high speed)
D Mode - 60 x 54 dpi (high resolution)

TIFF (TIFF.DRV) - This driver generates a Tagged Image File Format (TIFF) file. This file format can be read by several different types of programs. Thus, SCHEMA schematics can be read/displayed/printed by other programs which support this file format. Four resolutions can be chosen via the print mode as follows (horizontal by vertical):

A Mode - 300 x 300 dpi
B Mode - 150 x 150 dpi
C Mode - 100 x 100 dpi
D Mode - 75 x 75 dpi

Toshiba Wide-Carriage (ITOS351.DRV) - This driver supports the Toshiba P350 and P1351 wide-carriage printers. The printer is configured for the following print modes and dot resolutions (horizontal by vertical):

A Mode - 120 x 120 dpi
C Mode - 180 x 180 dpi
D Mode - 60 x 60 dpi

Toshiba Narrow-Carriage (ITOS340.DRV) - This driver supports the Toshiba P340 and P1340 narrow-carriage printers.

This printer driver has the same print resolutions as the Toshiba wide-carriage print driver.

E.6 PLOTTER

SCHEMA III supports several plotter types. The plotter type is defined in the configuration file with the **PLOTTER=** variable. Another variable, **PLOTC=**, is used to define the serial port used to communicate with the plotter. See Chapter 5, *SCHCFG*, for information on this variable. Only SCHPLOT uses these variables. Plotter drivers are part of the SCHPLOT program and thus, there are no individual plotter driver programs. All SCHEMA III plots require a single pen, i.e., Plotter Pen 1. Refer to Chapter 4, *SCHPLOT/SCHPRINT* for details regarding configuration variables and cable specs.

Appendix F. Object Libraries

F.1 THE STANDARD COMPONENT LIBRARY

The SCHEMA III Standard Library disks contains the following: ANALOG.LBR, ECL.LBR, CMOS.LBR, GRAPH.LBR, IEEE.LBR, MEMORY.LBR, MICRO.LBR, ROTATE.LBR, TTL1.LBR and TTL2.LBR.

A flexible naming convention has been used to organize SCHEMA III's component objects. All components consist of two or three types of objects called *Name*, *Body*, and *Nested* Objects (see Chapters 1 and 3).

Digital Name Objects use device nomenclature (such as 74LS00) as names while discrete component names begin with the letter D followed by a short abbreviation. For example: DRES (a resistor).

Body Object names consist of a prefix and suffix and are assigned according to type, physical orientation, and gate section. The Body Object prefix defines its type. The following type definitions are used:

A (Discrete components and analog ICs)
C (4000-Series CMOS)
E (ECL - 10K and 100K)
EPRMn (EPROM)
L (Line Driver/Receiver)
MEMnn (Memory ICs)
P (Microprocessor and peripheral ICs)
PRMnn (Bipolar PROM)
T (TTL - including ALS, AS, F, H, HC, L, LS, and S components).

Up to five alphanumeric characters can occupy the Body Object prefix (including the type designation). The Body Object suffix is always **-na**, where **n** is a numeric character denoting physical orientation and **a** is an alphabetical character defining gate section. Some sample Body Objects are:

T00-1A - Used to represent A-gate sections of 74ALS00, 74LS00, 74HC00, etc.
T00-2B - Used to represent B-gate sections of DeMorgan 74ALS00, 74LS00, 74HC00, etc.
P8048-1A - Used to represent an 8048 micoprocessor.

Nested Object names begin with the letter **O**. For example: ONAND.

Components are either digital (IC) or discrete objects. Digital components consist of a Name Object and one or more Body Objects.

The Name Object's name is the IC name, i.e., 74LS00, 74LS374, 74S258, etc. Each Name Object contains at least three fields: **DESC 1**, **DESC 2**, and **BODY** (the **DESC 3** and **DESC 4** fields are shipped undefined). The information in the **DESC** fields is used by the SCHPOST post-processors.

The **DESC 1** field contains the in-house (company) part number for the component. Omation has preassigned part numbers to each SCHEMA III component based on a part numbering system developed by Omation. The Omation part numbering system is:

XXX-YYYY-ZZZ

where **XXX** identifies a component family.

The component families are:

300 - TTL (74-, 74ALS-, 74AS-, 74F-, 74H-, 74L-, 74LS-, 74S-)
301 - CMOS (74HC-, 4000 Series)
302 - Analog (LM-)
303 - Bipolar PROM (AM-, HM-, MB-)
304 - EPROM/RAM
306 - Microprocessor/Peripheral
307 - 10K ECL
308 - 100K ECL
310 - Line Driver/Receiver/PAL
320 - TTL (54-, 54AS-, 54ALS-, 54F-, 54H-, 54HC-, 54L-, 54LS-)

YYYY identifies the component's generic number. For example, a 74LS10 is **0010**, a 4164 is **4164**, and so on.

ZZZ identifies a particular subgroup within a component family. These include:

001 - 74LS-, 4000 Series, Bipolar PROM, Analog,
Line Driver/Receiver, EPROM/RAM/PAL, Microprocessor/Peripheral
002 - 74S-, 74HC-
003 - 74-
004 - 74ALS-
005 - 74F-

For example:

300-0162-002 - 74S162
301-0162-002 - 74HC162
302-0555-001 - LM555
303-7131-001 - MB7131
304-6164-001 - 6164
306-8048-001 - 8048
310-1488-001 - MC1488

You may elect to follow Omation's part numbering system or change the **DESC 1** fields of each component's Name Object to reflect your company's part numbering system.

Up to 54 characters can be used in the **DESC 1** field.

The **DESC 2** field contains the component's name and a brief description of the component. This field may or may not require editing, depending on your requirements. Up to 54 characters can be used.

The **DESC 1** and **DESC 2** fields are only used by the SCHPOST post-processors. The **BODY** field contains the prefix (up to five characters) of the Body Object(s) that contains the visual information for the component. Most digital components have only one Body Object, i.e., T374-1A is the Body Object for a 74LS374. However, ICs with multiple sections have multiple Body Objects (T74-1A and T74-1B are the Body Objects for a 74LS74). In addition, gates have multiple Body Objects for each section and alternate DeMorgan-equivalent Body Objects, i.e., T00-1A, T00-1B, T00-1C, and T00-1D are the Body Objects for a 74LS00 (normal logic representation), while T00-2A, T00-2B, T00-2C, and T00-2D are the Body Objects for the DeMorgan representation of a 74LS00. Normally, there are no parameters associated with IC components. The only Schematic Editor **DRAW-OBJECT** Parameter String entries are the object's Name Object name and optional reference designator.

Discrete components are resistors, capacitors, transistors, switches, etc. As with ICs, each discrete component object consists of a Name Object and one or more Body Objects. All discrete Name Object names begin with the letter D. Like digital Name Objects, discrete component Name Objects contain at least three fields: **DESC 1**, **DESC 2**, and **BODY**. However, these fields are used differently from the equivalent IC component Name Object fields.

The **DESC 1** field is the in-house part number, but is normally expressed as **<P4>**. This informs the SCHPOST post-processors that the in-house part number, entered as the fourth **DRAW-OBJECT** Parameter String entry, should be used as the **DESC 1** field. The **DESC 2** field is the component's description, again taken from **DRAW-OBJECT** Parameter String entry. For example, a resistor's **DESC 2** field is: **RESISTOR,<P1>,**. Thus, the first **DRAW-OBJECT** Parameter String entry (value) is placed in the component's **DESC 2** field after **RESISTOR**. The **BODY** field contains the prefix of the Body Object(s) that contains the component's visual information. All discrete and analog component Body Object names begin with the letter **A**.

Discrete components have either two Body Object orientations (resistors, switches), or four Body Object orientations (transistors, diodes). Components with two Body Object orientations use Orientations 1 and 2 (ARES-1A, ARES-2A), while components with four Body Object orientations use Orientations 1, 2, 3, and 4 (ANPN-1A, ANPN-2A, ANPN-3A, ANPN-4A). A discrete and analog component's **DRAW-OBJECT** Parameter String contains the component's name, optional reference designator, at least one (three maximum) visual parameter (**PARM 1**, **PARM 2**, and **PARM 3**), and its in-house part number (**PARM 4**).

Components included with the libraries have been divided into tables, according to function. Tables F-1 through F-5 contain these objects. The tables are located at the end of this appendix. Although the list is not complete, the Browse commands in **DRAW-OBJECT** to print a list from the screen or **LIBRARY LOAD** in Chapter 3, *SCHDRAW* or DIR-OBJECT in Chapter 6, *SCHLIB* can be used to get a printer output of the components. The following listing is to be a guide for understanding the structure of the libraries.

NOTE

Due to the wide variety of voltages used, ECL and ANALOG parts, power pins and the net names given to them should be checked to assure that they are assigned to the net name desired.

Also, because the naming convention is limited to eight characters for 54 and 74, ALS and AS objects that are too long are abbreviated. For example: 5AL for 54ALS and 7AS for 74AS.

F.2 GRAPH.LBR: THE GRAPHICS LIBRARY

The SCHEMA III Libraries contain the GRAPH.LBR file (Graphics Library). This library contains general-purpose, non-component objects. These objects can be used to draw non-schematic drawings such as timing diagrams, block diagrams, flow charts, organizational charts, etc.

No special naming conventions have been used in the graphics library. Most objects are single-level objects, i.e., one Body Object (no separate Name Object). Only arrow objects (ARROW, BARROW) have both Name and Body Objects. Tables F-6 through F-9 contain the graphics objects, grouped according to the following functions: general graphic, assembly, flow chart, and timing.

F.3 IEEE.LBR: THE IEEE/IEC COMPONENT LIBRARY

The SCHEMA III IEEE/IEC Library disk contains the IEEE.LBR file. This library has IEEE/IEC symbols for some component objects contained in the Standard Libraries. Since the IEEE/IEC symbols have the same component names as the Standard Libraries, the libraries cannot be used in the same drawing or configuration file as the IEEE/IEC library. The IEEE/IEC symbols are used normally. For example, after the SCHDRAW **DRAW-OBJECT** command is selected, entering:

74LS00 <Enter>

retrieves the IEEE 74LS00 symbol instead of the conventional 74LS00 symbol. Tables F-10 through F-29 contain the IEEE/IEC symbols.

F.4 THE ROTATED / GATE PARTS

The SCHEMA III suplies rotated versions of all gated parts within their respective library. Since the rotated gates are simply additional orientations of existing gates, the ROTATED parts are used in conjunction with the Standard Library parts.

The ROTATED library parts supply Body Object -3x through -8x for all gates. As mentioned in paragraph F.1, Standard Libraries supplies Body Object orientations -1x and -2x, where orientation -1x is the normal representation (oriented left-to-right) and orientation -2x is the DeMorgan representation (oriented left-to-right). ROTATED library parts supplys the following additional orientations:

-3x (Normal, oriented bottom-to-top)
-4x (DeMorgan, oriented bottom-to-top)
-5x (Normal, oriented right-to-left)
-6x (DeMorgan, oriented right-to-left)
-7x (Normal, oriented top-to-bottom)
-8x (DeMorgan, oriented top-to-bottom)

For example, the right-to-left orientation, B-section Body Object for the 74LS00 is T00-5B, while the equivalent DeMorgan B-section Body Object is T00-6B. Tables F-30 contains the rotated gate Nested Objects.

Table F-1. Discrete Component Name Objects

DCAP	Capacitor (two orientations)
DCAPE	Electrolytic Capacitor (two orientations)
DCAPV	Variable Capacitor (two orientations)
DCHOKE	Choke (two orientations)
DCUR	SPICE Current Source
DDIODE	Diode (four orientations)
DIND	Inductor (two orientations)
DLED	LED (four orientations)
DNJFET	N-Channel JFET (four orientations)
DNOTE	SPICE Note
DNPN	NPN Transistor (eight orientations)
DPJFET	P-Channel JFET (four orientations)
DPNP	PNP Transistor (eight orientations)
DPOT	Variable Resistor (four orientations)
DRES	Resistor (two orientations)

Table F-1. Discrete Component Name Objects (Continued)

DRPAC8	8-Resistor Pack
DRPAC16	15 Common-Lead Resistor Pack
DSCR	SCR (four orientations)
DSGND	SPICE Ground Symbol
DSPDT	SPDT Switch (two orientations)
DSPST	SPST Switch (two orientations)
DVOLT	SPICE Voltage Source
DXFMR	Two-Winding, Non-Center Tapped Transformer (two orientations)
DXTAL	Crystal (two orientations)
DZENER	Zener Diode (four orientations)

Table F-2. Bipolar PROM Body Objects

PRM01-1A	32 X 8 Tri-State
PRM02-1A	32 X 8 Open Collector
PRM03-1A	256 X 4 Tri-State
PRM04-1A	256 X 4 Open Collector
PRM05-1A	512 X 4 Tri-State
PRM06-1A	512 X 4 Open Collector
PRM07-1A	512 X 8 Tri-State
PRM08-1A	512 X 8 Open Collector
PRM09-1A	512 X 8 Tri-State, Four Selects
PRM10-1A	512 X 8 Open Collector, Four Selects
PRM11-1A	1024 X 4 Tri-State
PRM12-1A	1024 X 4 Open Collector
PRM13-1A	1024 X 8 Tri-State
PRM14-1A	1024 X 8 Open Collector
PRM15-1A	2048 X 4 Tri-State
PRM16-1A	2048 X 4 Open Collector
PRM17-1A	2048 X 8 Tri-State

Table F-2. Bipolar PROM Body Objects (Continued)

PRM18-1A	2048 X 8 Open Collector
PRM19-1A	4096 X 4 Tri-State
PRM20-1A	4096 X 4 Open Collector
PRM21-1A	4096 X 8 Tri-State, Registered
PRM22-1A	512 X 8 Tri-State, Registered
PRM23-1A	1024 X 8 Tri-State, Registered
PRM24-1A	2048 X 8 Tri-State, Registered

Table F-3. EPROM/RAM/PAL Body Objects

EPRM1-1A	2K X 8 EPROM
EPRM2-1A	4K X 8 EPROM
EPRM3-1A	8K X 8 EPROM
EPRM4-1A	16K X 8 EPROM
EPRM5-1A	32K X 8 EPROM
MEM1-1A	2K X 8 RAM
MEM2-1A	8K X 8 RAM
MEM3-1A	64K DRAM
MEM4-1A	256K DRAM
PAL1-1A	PAL16L8
PAL2-1A	PAL16R4
PAL3-1A	PAL16R6
PAL4-1A	PAL16R8
PAL5-1A	PAL20R4
PAL6-1A	PAL20R6
PAL7-1A	PAL20R8
PAL8-1A	PAL20L8

Table F-4. Nested Objects

OAND	2-Input AND Gate
OANDD	2-Input AND Gate, DeMorgan
OAND3	3-Input AND Gate
OAND3D	3-Input AND Gate, DeMorgan
OAND4	4-Input AND Gate
OAND4D	4-Input AND Gate, DeMorgan
OBUF	Non-Inverting Buffer
OBUFD	Non-Inverting Buffer, DeMorgan
ODFF	D Flip-Flop With PRESET And CLEAR
OINV	Inverting Buffer
ONIVD	Inverting Buffer, DeMorgan
OINVS	Inverting Buffer, Schmitt Trigger
OINVSD	Inverting Buffer, Schmitt Trigger, DeMorgan
OJKFF	J-K Flip-Flop With PRESET And CLEAR
ONAND	2-Input NAND Gate
ONANDD	2-Input NAND Gate, DeMorgan
ONAND3	3-Input NAND Gate

Table F-4. Nested Objects (Continued)

ONAND3D	3-Input NAND Gate, DeMorgan
ONAND4	4-Input NAND Gate
ONAND4D	4-Input NAND Gate, DeMorgan
ONAND4DS	4-Input NAND Gate, Schmitt Trigger, DeMorgan
ONOR	2-Input NOR Gate
ONORD	2-Input NOR Gate, DeMorgan
ONOR3	3-Input NOR Gate
ONOR3D	3-Input NOR Gate, DeMorgan
OOR	2-Input OR Gate
OORD	2-Input OR Gate, DeMorgan
OTSBUF	Non-Inverting Buffer With Active High Enable
OTSBUFL	Non-Inverting Buffer With Active Low Enable
OVCO	VCO
OVCOEN	VCO With Enable
OXOR	2-Input Exclusive OR Gate

Table F-5. Non-Component Objects

AR1	Thin Arc, 12 to 3 O'Clock
ARB1	Fat Arc, 12 to 3 O'Clock
AR2	Thin Arc, 3 to 6 O'Clock
ARB2	Fat Arc, 3 to 6 O'Clock
AR3	Thin Arc, 6 to 9 O'Clock
ARB3	Fat Arc, 6 to 9 O'Clock
AR4	Thin Arc, 9 to 12 O'Clock
ARB4	Fat Arc, 9 to 12 O'Clock
DOT	Connection Dot
GND	Digital Ground
GNDA	Analog Ground
GNDC	Chassis Ground
OMATION1	Omation Logo
PIHDR1	Pin Header, Horizontal, No Bubble
PIHDR2	Pin Header, Horizontal, Right Bubble
PIHDR3	Pin Header, Horizontal, Left Bubble

Table F-5. Non-Component Objects (Continued)

PIHDR4	Pin Header, Vertical, No Bubble
PIHDR5	Pin Header, Vertical, Bottom Bubble
PIHDR6	Pin Header, Vertical, Top Bubble

Table F-6. GRAPH.LBR General Graphics Objects

PAT1

PAT2

PAT3

PAT4

PAT5

PAT6

PAT7

PAT8

PAT9

PAT10

Table F-6. GRAPH.LBR General Graphics Objects (Continued)

OARW-1A

OARW-2A

OARW-3A

OARW-4A

OARW-1B

OARW-2B

OARW-3B

OARW-4B

OARW-1C

OARW-2C

OARW-3C

OARW-4C

OARW-1D

OARW-2D

OARW-3D

Table F-6. GRAPH.LBR General Graphics Objects (Continued)

	OARW-4D	
BARROW	Large Arrow Name Object With The Following Body Objects:	
	OBARW-1A	
	OBARW-2A	
	OBARW-3A	
	OBARW-4A	
	OBARW-1B	
	OBARW-2B	
	OBARW-3B	
	OBARW-4B	
	OBARW-1C	
	OBARW-2C	
	OBARW-3C	
	OBARW-4C	

Table F-6. GRAPH.LBR General Graphics Objects (Continued)

OBARW-1D	
OBARW-2D	
OBARW-3D	
OBARW-4D	
CIRC8	8-Pixel Diameter Circle
CIRC16	16-Pixel Diameter Circle
CIRC32	32-Pixel Diameter Circle
CIRC48	48-Pixel Diameter Circle
CIRC64	64-Pixel Diameter Circle

Table F-7. GRAPH.LBR Assembly Drawing Objects

LDOT	Single Pad Outline
LRESR	Resistor Outline, Horizontal
LRES	Resistor Outline, Vertical
LCAP1R	Small Capacitor Outline, Horizontal
LCAP2R	Medium Capacitor Outline, Horizontal
LCAP3R	Large Capacitor Outline, Horizontal
LCAP1	Small Capacitor Outline, Vertical
LCAP2	Medium Capacitor Outline, Vertical
LCAP3	Large Capacitor Outline, Vertical
LEURO	96-Pin EURO Connector Outline
LIC14	14-Pin IC Outline, Vertical
LIC14R	14-Pin IC Outline, Horizontal

Table F-7. GRAPH.LBR Assembly Drawing Objects (Continued)

LIC16	16-Pin IC Outline, Vertical
LIC16R	16-Pin IC Outline, Horizontal
LIC18	18-Pin IC Outline, Vertical
LIC18R	18-Pin IC Outline, Horizontal
LIC20	20-Pin IC Outline, Vertical
LIC20R	20-Pin IC Outline, Horizontal
LIC24	24-Pin IC Outline, Vertical
LIC24R	24-Pin IC Outline, Horizontal
LIC28	28-Pin IC Outline, Vertical
LIC28R	28-Pin IC Outline, Horizontal
LIC40	40-Pin IC Outline, Vertical
LIC40R	40-Pin IC Outline, Horizontal
LXSTR	Transistor Outline

Table F-8. GRAPH.LBR Flow Chart Objects

BOX1	Rectangle With Single Text Line
BOX2	Rectangle With Two Text Lines
BOX3	Rectangle With Three Text Lines
BOX4	Rectangle With Four Text Lines
COND1	Small Conditional Block, Two Text Lines
COND2	Medium Conditional Block, Two Text Lines
COND3	Large Conditional Block, Two Text Lines
LABEL	Label Block

Table F-9. GRAPH.LBR Timing Diagram Objects

CLK1	Single Clock Pulse
CLK2	Two Clock Pulses
CLK4	Four Clock Pulses
CLK8	Eight Clock Pulses
PULSEM0	Diagonal Single Line, Downward Slope
PULSEM1	Single Clock Width, Negative Pulse
PULSEM2	Two Clock Widths, Negative Pulse
PULSEM3	Three Clock Widths, Negative Pulse
PULSEM4	Four Clock Widths, Negative Pulse
PULSEP0	Diagonal Single Line, Upward Slope
PULSEP1	Single Clock Width, Positive Pulse

Table F-9. GRAPH.LBR Timing Diagram Objects (Continued)

PULSEP2	Two Clock Widths, Positive Pulse
PULSEP3	Three Clock Widths, Positive Pulse
PULSEP4	Four Clock Widths, Positive Pulse

Table F-10. IEEE.LBR 74-Series Name Objects

7406	7407	7416	7417
74121	74180		

Table F-11. IEEE.LBR 74ALS-Series Name Objects

74ALS00	74ALS02	74ALS03	74ALS04	74ALS05	74ALS08
74ALS09	74ALS10	74ALS11	74ALS12	74ALS15	74ALS20
74ALS21	74ALS22	74ALS27	74ALS28	74ALS30	74ALS32
74ALS33	74ALS37	74ALS38	74ALS40	74ALS74	74ALS86
74ALS109	74ALS112	74ALS133	74ALS137	74ALS138	74ALS139
74ALS151	74ALS153	74ALS157	74ALS158	74ALS160	74ALS161
74ALS162	74ALS163	74ALS164	74ALS165	74ALS166	74ALS168
74ALS169	74ALS174	74ALS175	74ALS190	74ALS191	74ALS192
74ALS193	74ALS240	74ALS241	74ALS242	74ALS243	74ALS244
74ALS245	74ALS251	74ALS253	74ALS257	74ALS258	74ALS259
74ALS273	74ALS299	74ALS352	74ALS353	74ALS365	74ALS366
74ALS367	74ALS368	74ALS373	74ALS374	74ALS533	74ALS534
74ALS540	74ALS541	74ALS563	74ALS564	74ALS573	74ALS574
74ALS580	74ALS620	74ALS621	74ALS622	74ALS623	74ALS638
74ALS639	74ALS640	74ALS641	74ALS642	74ALS643	74ALS644
74ALS645					

Table F-12. IEEE.LBR 74F-Series Name Objects

74F00	74F02	74F04	74F08	74F10	74F11
74F20	74F32	74F74	74F86	74F109	74F112
74F138	74F139	74F148	74F151	74F153	74F157
74F158	74F160	74F161	74F162	74F163	74F164
74F168	74F169	74F174	74F175	74F181	74F190
74F191	74F192	74F193	74F240	74F241	74F242
74F243	74F244	74F245	74F251	74F253	74F257
74F258	74F280	74F283	74F299	74F352	74F353
74F373	74F374	74F378	74F379	74F533	74F534

Table F-13. IEEE.LBR 74HC-Series Name Objects

74HC00	74HC02	74HC03	74HC04	74HC08	74HC10
74HC11	74HC14	74HC20	74HC21	74HC27	74HC30
74HC32	74HC51	74HC74	74HC85	74HC86	74HC90
74HC92	74HC93	74HC107	74HC109	74HC112	74HC123
74HC125	74HC126	74HC133	74HC137	74HC138	74HC139
74HC151	74HC153	74HC157	74HC158	74HC160	74HC161
74HC162	74HC163	74HC164	74HC165	74HC166	74HC173
74HC174	74HC175	74HC181	74HC190	74HC191	74HC192
74HC193	74HC221	74HC240	74HC241	74HC242	74HC243
74HC244	74HC245	74HC251	74HC253	74HC257	74HC258
74HC259	74HC273	74HC280	74HC283	74HC298	74HC299
74HC352	74HC353	74HC365	74HC366	74HC367	74HC368
74HC373	74HC374	74HC377	74HC378	74HC379	74HC390
74HC393	74HC533	74HC534	74HC540	74HC541	74HC563
74HC564	74HC573	74HC574	74HC640	74HC643	74HC670
74HC4015	74HC4020	74HC4040	74HC4049	74HC4050	74HC4051
74HC4052	74HC4053	74HC4066			

Table F-14. IEEE.LBR 74LS-Series Name Objects

74LS00	74LS02	74LS03	74LS04	74LS05	74LS08
74LS09	74LS10	74LS11	74LS12	74LS13	74LS14
74LS15	74LS18	74LS19	74LS20	74LS21	74LS22
74LS24	74LS27	74LS28	74LS30	74LS32	74LS33
74LS37	74LS38	74LS40	74LS42	74LS51	74LS54
74LS74	74LS83	74LS85	74LS86	74LS90	74LS92
74LS93	74LS107	74LS109	74LS112	74LS122	74LS123
74LS124	74LS125	74LS126	74LS132	74LS137	74LS138
74LS139	74LS145	74LS148	74LS151	74LS153	74LS155
74LS156	74LS157	74LS158	74LS160	74LS161	74LS162
74LS163	74LS164	74LS165	74LS166	74LS168	74LS169
74LS170	74LS173	74LS174	74LS175	74LS181	74LS190
74LS191	74LS192	74LS193	74LS221	74LS240	74LS241
74LS242	74LS243	74LS244	74LS245	74LS251	74LS253
74LS257	74LS258	74LS259	74LS273	74LS280	74LS283
74LS290	74LS293	74LS298	74LS299	74LS324	74LS348
74LS352	74LS353	74LS363	74LS364	74LS365	74LS366
74LS367	74LS368	74LS373	74LS374	74LS376	74LS377
74LS378	74LS379	74LS381	74LS390	74LS393	74LS422
74LS423	74LS540	74LS541	74LS620	74LS621	74LS622
74LS623	74LS624	74LS625	74LS626	74LS627	74LS629
74LS638	74LS639	74LS640	74LS641	74LS642	74LS643
74LS644	74LS645	74LS670			

Table F-15. IEEE.LBR 74S-Series Name Objects

74S00	74S02	74S03	74S04	74S05	74S08
74S09	74S10	74S11	74S15	74S20	74S22
74S30	74S32	74S37	74S38	74S40	74S74
74S85	74S86	74S112	74S124	74S132	74S133
74S134	74S138	74S139	74S151	74S153	74S157
74S158	74S162	74S163	74S168	74S169	74S174
74S175	74S181	74S182	74S240	74S241	74S244
74S251	74S257	74S258	74S280	74S283	74S299
74S373	74S374	74S381			

Table F-16. IEEE.LBR 4000-Series CMOS Name Objects

4001	4008	4011	4012	4013	4014
4015	4019	4020	4021	4023	4025
4027	4028	4040	4049	4050	4051
4052	4069	4070	4071	4076	4081
4164	4519	4539	40014	40085	40097
40098	40161	40163	40174	40175	40193
41256					

Table F-17. IEEE.LBR Bipolar PROM Name Objects

AM27S12	AM27S13	AM27S18	AM27S19	AM27S20	AM27S21
AM27S25	AM27S28	AM27S29	AM27S30	AM27S31	AM27S32
AM27S33	AM27S35	AM27S37	AM27S40	AM27S41	AM27S43
AM27S45	AM27S47	AM26LS31	AM26LS32	AM26LS33	AM27S180
AM27S181	AM27S184	AM27S185	AM27S190	AM27S191	

HM7602	HM7603	HM7610	HM7611	HM7620	HM7621
HM7640	HM7641	HM7642	HM7643	HM7648	HM7649
HM7680	HM7681	HM7684	HM7685	HM76160	HM76161
HM76164	HM76165				

MB7121	MB7122	MB7123	MB7124	MB7127	MB7128
MB7131	MB7132	MB7134	MB7137	MB7138	MB7142

Table F-18. IEEE.LBR Analog and Line Driver/Receiver Name Objects

LM555	LM556			
MC1488	MC1489	MC3486	MC3487	
75172	75173	75176	75188	75189

Table F-19. IEEE.LBR EPROM/RAM/PAL Name Objects

2716	2732	2764	27128
27256	4164	41256	6116
6164			

Table F-20. IEEE.LBR Microprocessor and Peripheral Name Objects

2901	2903	2911	6801
6803	6809	6845	8031
8035	8039	8040	8048
8049	8050	8051	8086
8087	8088	8237	8250
8251	8253	8254	8255
8259	8272	8284	8288
8748	8749	8751	16201
29203	32016	32032	32082
68000	68020	68451	68881
80186	80188	80286	80287
82284	82288	UPD765	Z80CPU
Z80CTC	Z80PIO	Z80SIO	

Table F-21. IEEE.LBR Discrete Component Name Objects

DCAP	Capacitor (two orientations)
DCAPE	Electrolytic Capacitor (two orientations)
DCAPV	Variable Capacitor (two orientations)
DCHOKE	Choke (two orientations)
DCUR	SPICE Current Source
DDIODE	Diode (four orientations)
DIND	Inductor (two orientations)
DLED	LED (four orientations)
DNJFET	N-Channel JFET (four orientations)
DNOTE	SPICE Note
DNPN	NPN Transistor (eight orientations)
DPJFET	P-Channel JFET (four orientations)
DPNP	PNP Transistor (eight orientations)
DPOT	Variable Resistor (four orientations)
DRES	Resistor (two orientations)

Table F-21. IEEE.LBR Discrete Component Name Objects (Continued)

DRPAC8	8-Resistor Pack
DRPAC16	15 Common-Lead Resistor Pack
DSCR	SCR (four orientations)
DSGND	SPICE Ground Symbol
DSPDT	SPDT Switch (two orientations)
DSPST	SPST Switch (two orientations)
DVOLT	SPICE Voltage Source
DXFMR	Two-Winding, Non-Center Tapped Transformer (two orientations)
DXTAL	Crystal (two orientations)
DZENER	Zener Diode (four orientations)

Table F-22. IEEE.LBR 74-, 74ALS-, 74F-, 74HC-, 74LS-, and 74S-Series Body Objects

	T00-1A	T00-1B	T00-1C	T00-1D	T00-2A
T00-2B	T00-2C	T00-2D	T02-1A	T02-1B	T02-1C
T02-1D	T02-2A	T02-2B	T02-2C	T02-2D	T03-1A
T03-1B	T03-1C	T03-1D	T03-2A	T03-2B	T03-2C
T03-2D	T04-1A	T04-1B	T04-1C	T04-1D	T04-1E
T04-1F	T04-2A	T04-2B	T04-2C	T04-2D	T04-2E
T04-2F	T06-1A	T06-1B	T06-1C	T06-1D	T06-1E
T06-1F	T06-2A	T06-2B	T06-2C	T06-2D	T06-2E
T06-2F	T07-1A	T07-1B	T07-1C	T07-1D	T07-1E
T07-1F	T07-2A	T07-2B	T07-2C	T07-2D	T07-2E
T07-2F	T08-1A	T08-1B	T08-1C	T08-1D	T08-2A
T08-2B	T08-2C	T08-2D	T09-1A	T09-1B	T09-1C
T09-1D	T09-2A	T09-2B	T09-2C	T09-2D	T10-1A
T10-1B	T10-1C	T10-2A	T10-2B	T10-2C	T11-1A
T11-1B	T11-1C	T11-2A	T11-2B	T11-2C	T12-1A
T12-1B	T12-1C	T12-2A	T12-2B	T12-2C	T13-1A
T13-1B	T13-2A	T13-2B	T14-1A	T14-1B	T14-1C
T14-1D	T14-1E	T14-1F	T14-2A	T14-2B	T14-2C
T14-2D	T14-2E	T14-2F	T15-1A	T15-1B	T15-1C
T15-2A	T15-2B	T15-2C	T18-1A	T18-1B	T18-2A
T18-2B	T20-1A	T20-1B	T20-2A	T20-2B	T21-1A
T21-1B	T21-2A	T21-2B	T22-1A	T22-1B	T22-2A
T22-2B	T24-1A	T24-1B	T24-1C	T24-1D	T24-2A
T24-2B	T24-2C	T24-2D	T27-1A	T27-1B	T27-1C
T27-2A	T27-2B	T27-2C	T30-1A	T30-2A	T32-1A
T32-1B	T32-1C	T32-1D	T32-2A	T32-2B	T32-2C
T32-2D	T33-1A	T33-1B	T33-1C	T33-1D	T33-2A
T33-2B	T33-2C	T33-2D	T42-1A	T51-1A	T51-1B
T54-1A	T74-1A	T74-1B	T83-1A	T85-1A	T86-1A
T86-1B	T86-1C	T86-1D	T90-1A	T92-1A	T93-1A

Table F-22. IEEE.LBR 74-, 74ALS-, 74F-, 74HC-, 74LS-, and 74S-Series Body Objects (Continued)

T107-1A	T107-1B	T109-1A	T109-1B	T112-1A	T112-1B
T121-1A	T122-1A	T122-1B	T123-1A	T123-1B	T124-1A
T124-1B	T125-1A	T125-1B	T125-1C	T125-1D	T126-1A
T126-1B	T126-1C	T126-1D	T132-1A	T132-1B	T132-1C
T132-1D	T132-2A	T132-2B	T132-2C	T132-2D	T133-1A
T134-1A	T137-1A	T138-1A	T139-1A	T139-1B	T148-1A
T151-1A	T153-1A	T155-1A	T156-1A	T157-1A	T158-1A
T160-1A	T161-1A	T162-1A	T163-1A	T164-1A	T165-1A
T166-1A	T168-1A	T169-1A	T170-1A	T173-1A	T174-1A
T175-1A	T180-1A	T181-1A	T182-1A	T190-1A	T191-1A
T192-1A	T193-1A	T221-1A	T221-1B	T240-1A	T240-1B
T241-1A	T241-1B	T242-1A	T243-1A	T244-1A	T244-1B
T245-1A	T251-1A	T253-1A	T257-1A	T258-1A	T259-1A
T273-1A	T280-1A	T283-1A	T290-1A	T293-1A	T298-1A
T299-1A	T324-1A	T348-1A	T352-1A	T353-1A	T363-1A
T364-1A	T365-1A	T366-1A	T367-1A	T368-1A	T373-1A
T374-1A	T377-1A	T378-1A	T379-1A	T381-1A	T390-1A
T390-1B	T393-1A	T393-1B	T533-1A	T534-1A	T540-1A
T541-1A	T563-1A	T564-1A	T573-1A	T574-1A	T620-1A
T623-1A	T638-1A	T643-1A	T670-1A		

Table F-23. IEEE.LBR 4000-Series CMOS Body Objects

PRM01-1A	32 X 8 Tri-State
PRM02-1A	32 X 8 Open Collector
PRM03-1A	256 X 4 Tri-State
PRM04-1A	256 X 4 Open Collector
PRM05-1A	512 X 4 Tri-State
PRM06-1A	512 X 4 Open Collector
PRM07-1A	512 X 8 Tri-State
PRM08-1A	512 X 8 Open Collector
PRM09-1A	512 X 8 Tri-State, Four Selects
PRM10-1A	512 X 8 Open Collector, Four Selects
PRM11-1A	1024 X 4 Tri-State
PRM12-1A	1024 X 4 Open Collector
PRM13-1A	1024 X 8 Tri-State
PRM14-1A	1024 X 8 Open Collector
PRM15-1A	2048 X 4 Tri-State
PRM16-1A	2048 X 4 Open Collector
PRM17-1A	2048 X 8 Tri-State

Table F-24. IEEE.LBR Bipolar PROM Body Objects

PRM01-1A	32 X 8 Tri-State
PRM02-1A	32 X 8 Open Collector
PRM03-1A	256 X 4 Tri-State
PRM04-1A	256 X 4 Open Collector
PRM05-1A	512 X 4 Tri-State
PRM06-1A	512 X 4 Open Collector
PRM07-1A	512 X 8 Tri-State
PRM08-1A	512 X 8 Open Collector
PRM09-1A	512 X 8 Tri-State, Four Selects
PRM10-1A	512 X 8 Open Collector, Four Selects
PRM11-1A	1024 X 4 Tri-State
PRM12-1A	1024 X 4 Open Collector
PRM13-1A	1024 X 8 Tri-State
PRM14-1A	1024 X 8 Open Collector
PRM15-1A	2048 X 4 Tri-State
PRM16-1A	2048 X 4 Open Collector
PRM17-1A	2048 X 8 Tri-State

Table F-24. IEEE.LBR Bipolar PROM Body Objects (Continued)

PRM18-1A	2048 X 8 Open Collector
PRM19-1A	4096 X 4 Tri-State
PRM20-1A	4096 X 4 Open Collector
PRM21-1A	4096 X 8 Tri-State, Registered
PRM22-1A	512 X 8 Tri-State, Registered
PRM23-1A	1024 X 8 Tri-State, Registered
PRM24-1A	2048 X 8 Tri-State, Registered

Table F-25. IEEE.LBR Discrete/Analog and Line Driver/Receiver Component Body Objects

A555-1A	A556-1A	A556-1B	ACAP-1A		
ACAP-2A	ACHK-1A	ACHK-2A	ADIO-1A	ADIO-2A	ADIO-3A
ADIO-4A	AIND-1A	AIND-2A	ALED-1A	ALED-2A	ALED-3A
ALED-4A					
ANJF-1A	ANJF-2A	ANJF-3A			
ANJF-4A	ANPN-1A	ANPN-2A	ANPN-3A	ANPN-4A	APJF-1A
APJF-2A	APJF-3A	APJF-4A	APNP-1A	APNP-2A	APNP-3A
APNP-4A	APOT-1A	APOT-2A	APOT-3A	APOT-4A	ARES-1A
ARES-2A	ASCR-1A	ASCR-2A	ASCR-3A	ASCR-4A	ACAPE-1A
ACAPE-2A					
ARP08-1A	ARP16-1A				
ASPDT-1A	ASPDT-2A	ASPST-1A	ASPST-2A	AVCAP-1A	AVCAP-2A
AXFMR-1A	AXFMR-2A	AXTAL-1A	AXTAL-2A	AZDIO-1A	AZDIO-2A
AZDIO-3A	AZDIO-4A				
L174-1B	L175-1B	L176-1A	L188-1A		
L188-1B	L188-1C	L188-1D	L189-1A	L189-1B	L189-1C
L189-1D	L2631-1A	L2632-1A	L2633-1A		

Table F-26. IEEE.LBR EPROM/RAM/PAL Body Objects

EPRM1-1A	2K X 8 EPROM
EPRM2-1A	4K X 8 EPROM
EPRM3-1A	8K X 8 EPROM
EPRM4-1A	16K X 8 EPROM
EPRM5-1A	32K X 8 EPROM
MEM1-1A	2K X 8 RAM
MEM2-1A	8K X 8 RAM
MEM3-1A	64K DRAM
MEM4-1A	256K DRAM
PAL1-1A	PAL16L8
PAL2-1A	PAL16R4
PAL3-1A	PAL16R6
PAL4-1A	PAL16R8
PAL5-1A	PAL20R4
PAL6-1A	PAL20R6
PAL7-1A	PAL20R8
PAL8-1A	PAL20L8

Table F-27. IEEE.LBR Microprocessor and Peripheral Body Objects

P0186-1A
P0188-1A
P0286-1A
P0286-2A
P0287-1A
P2016-1A
P2032-1A
P2082-1A
P2284-1A
P2288-1A
P2901-1A
P2903-1A
P2911-1A
P6201-1A
P6800-1A
P6801-1A
P6809-1A
P6845-1A
P8020-1A
P8048-1A
P8051-1A
P8086-1A
P8087-1A
P8088-1A
P8237-1A
P8250-1A
P8251-1A
P8253-1A
P8255-1A
P8259-1A
P8272-1A
P8284-1A
P8288-1A
P8451-1A
P8881-1A
PZ80-1A
PCTC-1A
PPIO-1A
PSIO-1A

Table F-28. IEEE.LBR Nested Objects

OAND	2-Input AND Gate
OANDD	2-Input AND Gate, DeMorgan
OAND3	3-Input AND Gate
OAND3D	3-Input AND Gate, DeMorgan
OAND4	4-Input AND Gate
OAND4D	4-Input AND Gate, DeMorgan
OBUF	Non-Inverting Buffer
OBUFD	Non-Inverting Buffer, DeMorgan
ODFF	D Flip-Flop With PRESET And CLEAR
OINV	Inverting Buffer
ONIVD	Inverting Buffer, DeMorgan
OINVS	Inverting Buffer, Schmitt Trigger
OINVSD	Inverting Buffer, Schmitt Trigger, DeMorgan
OJKFF	J-K Flip-Flop With PRESET And CLEAR
ONAND	2-Input NAND Gate
ONANDD	2-Input NAND Gate, DeMorgan
ONAND3	3-Input NAND Gate

Table F-28. IEEE.LBR Nested Objects (Continued)

ONAND3D	3-Input NAND Gate, DeMorgan
ONAND4	4-Input NAND Gate
ONAND4D	4-Input NAND Gate, DeMorgan
ONAND4DS	4-Input NAND Gate, Schmitt Trigger, DeMorgan
ONOR	2-Input NOR Gate
ONORD	2-Input NOR Gate, DeMorgan
ONOR3	3-Input NOR Gate
ONOR3D	3-Input NOR Gate, DeMorgan
OOR	2-Input OR Gate
OORD	2-Input OR Gate, DeMorgan
OTSBUF	Non-Inverting Buffer With Active High Enable
OTSBUFL	Non-Inverting Buffer With Active Low Enable
OVCO	VCO
OVCOEN	VCO With Enable
OXOR	2-Input Exclusive OR Gate

Table F-29. IEEE.LBR Non-Component Objects

AR1	Thin Arc, 12 to 3 O'Clock
ARB1	Fat Arc, 12 to 3 O'Clock
AR2	Thin Arc, 3 to 6 O'Clock
ARB2	Fat Arc, 3 to 6 O'Clock
AR3	Thin Arc, 6 to 9 O'Clock
ARB3	Fat Arc, 6 to 9 O'Clock
AR4	Thin Arc, 9 to 12 O'Clock
ARB4	Fat Arc, 9 to 12 O'Clock
DOT	Connection Dot
GND	Digital Ground
GNDA	Analog Ground
GNDC	Chassis Ground
OMATION1	Omation Logo
PIHDR1	Pin Header, Horizontal, No Bubble

Table F-29. IEEE.LBR Non-Component Objects (Continued)

PIHDR2	Pin Header, Horizontal, Right Bubble
PIHDR3	Pin Header, Horizontal, Left Bubble
PIHDR4	Pin Header, Vertical, No Bubble
PIHDR5	Pin Header, Vertical, Bottom Bubble
PIHDR6	Pin Header, Vertical, Top Bubble

Table F-30. ROTATED Nested Objects

O	Omation Letter "O" Logo Graphics
A2	4-Input, 270-Degree NOR Gate, DeMorgan
A9	4-Input, 90-Degree NOR Gate, DeMorgan
B1	180-Degree Inverting Buffer, DeMorgan
E2	4-Input, 270-Degree OR Gate, DeMorgan
E9	4-Input, 90-Degree OR Gate, DeMorgan
I2	270-Degree Inverting Buffer, DeMorgan
I9	90-Degree Inverting Buffer, DeMorgan
MN	Non-Inverting Buffer, DeMorgan
M1	Non-Inverting Buffer, 180-Degree, DeMorgan
M2	Non-Inverting Buffer, 270-Degree, DeMorgan
M9	Non-Inverting Buffer, 90-Degree, DeMorgan
N1	4-Input, 180-Degree NAND Gate, DeMorgan
N2	4-Input, 270-Degree NAND Gate, DeMorgan
N9	4-Input, 90-Degree NAND Gate, DeMorgan
Q1	3-Input, 180-Degree NOR Gate, DeMorgan
Q2	3-Input, 270-Degree NOR Gate, DeMorgan
Q9	3-Input, 90-Degree NOR Gate, DeMorgan
S0	13-Input, NAND Gate, DeMorgan
S1	13-Input, 180-Degree NAND Gate, DeMorgan
S2	13-Input, 270-Degree NAND Gate, DeMorgan
S9	13-Input, 90-Degree NAND Gate, DeMorgan

Table F-30. ROTATED Nested Objects (Continued)

W1	8-Input, 180-Degree NAND Gate, DeMorgan
W2	8-Input, 270-Degree NAND Gate, DeMorgan
W9	8-Input, 90-Degree NAND Gate, DeMorgan
X1	3-Input, 180-Degree OR Gate, DeMorgan
X2	3-Input, 270-Degree OR Gate, DeMorgan
X9	3-Input, 90-Degree OR Gate, DeMorgan
Z1	8-Input, 180-Degree NOR Gate, DeMorgan
Z2	8-Input, 270-Degree NOR Gate, DeMorgan
Z9	8-Input, 90-Degree NOR Gate, DeMorgan
NN1	4-Input, 180-Degree AND Gate, DeMorgan
NN2	4-Input, 270-Degree AND Gate, DeMorgan
NN9	4-Input, 90-Degree AND Gate, DeMorgan
OSM	Schmitt Trigger
RA1	4-Input, 180-Degree NOR Gate, DeMorgan
RE1	4-Input, 180-Degree OR Gate, DeMorgan
RN1	4-Input, 180-Degree NAND Gate, DeMorgan With Shifted Parameters
RW1	8-Input, 180-Degree NAND Gate, DeMorgan With Shifted Parameters
BUF9	90-Degree, Non-Inverting Buffer, DeMorgan

Table F-30. ROTATED Nested Objects (Continued)

OAOI	AND-OR-INVERT Gate
ORES	Horizontal Resistor
OA090	90-Degree AND Gate
OA180	180-Degree AND Gate
OA270	270-Degree AND Gate
ON000	Arc, 1 Visual Grid Unit
ON090	90-Degree Arc, 1 Visual Grid Unit
ON180	180-Degree Arc, 1 Visual Grid Unit
ON270	270-Degree Arc, 1 Visual Grid Unit
OOREN	OR Gate With 1 Inverting and Non-Inverting Input, and No Output
OORARC	Arc, 1 Visual Grid Unit
OOREN1	OR Gate With 1 Inverting and Non-Inverting Input, and 1 Non-Inverting Output
OOR090	90-Degree OR Gate With No Inputs
OOR180	180-Degree OR Gate With No Inputs
OOR270	270-Degree OR Gate With No Inputs
OTX000	Transmission Gate
OTX090	90-Degree Transmission Gate
ROR180	180-Degree OR Gate With No Inputs and Shifted Parameters
RTX000	Transmission Gate With Shifted Parameters

Table F-30. ROTATED Nested Objects (Continued)

MBUF090	90-Degree Buffer With No Inputs
MBUF180	180-Degree Buffer With No Inputs
MBUF270	270-Degree Buffer With No Inputs
MINV090	90-Degree Inverter With No Inputs
MINV180	180-Degree Inverter With No Inputs
MINV270	270-Degree Inverter With no Inputs
MNOR180	180-Degree NOR Gate With No Inputs and Shifted Parameters
OAND090	90-Degree AND Gate With No Inputs
OAND180	180-Degree AND Gate With No Inputs
OAND270	270-Degree AND Gate With No Inputs
OAOI090	90-Degree AND-OR-INVERT With No Inputs
OAOI180	180-Degree AND-OR-INVERT With No Inputs
OAOI270	270-Degree AND-OR-INVERT With No Inputs
OBUF090	90-Degree Non-Inverting Buffer
OBUF180	180-Degree Non-Inverting Buffer
OBUF270	270-Degree Non-Inverting Buffer
OINV090	90-Inverter
OINV180	180-Degree Inverter
OINV270	270-Degree Inverter
ONOR090	90-Degree NOR Gate With No Inputs
ONOR180	180-Degree NOR Gate With No Inputs
ONOR270	270-Degree NOR Gate With No Inputs

Table F-30. ROTATED Nested Objects (Continued)

OXIN090	90-Degree Arc, 2 Visual Grid Units
OXIN180	180-Degree Arc, 2 Visual Grid Units
OXIN270	270-Degree Arc, 2 Visual Grid Units
RAND180	180-Degree AND Gate With No Inputs and Shifted Parameters
RNOR180	180-Degree NOR Gate With No Inputs and Shifted Parameters
ONAND090	90-Degree NAND Gate With No Inputs
ONAND180	180-Degree NAND Gate With No Inputs
ONAND270	270-Degree NAND Gate With No Inputs
OSCHMITT	Schmitt Trigger
RNAND180	180-Degree NAND Gate With No Inputs and Shifted Parameters

INDEX

A

Ampersand variables
Manager, 2-38
Apple Imagewriter, E-13
Arc, 3-52
Archive a Project, 1-146,6-10
ASCII IN / ASCII OUT LIBRARY 1, 1-145,6-7,**6-11**
ASCII IN NAME OBJECTS TO LIB 1, **6-11**
ASCII IN SYNTAX CHECK, **6-12**
ASCII OUT NAME OBJECTS LIB 1, **6-13**
Assembly Drawing Objects, F-21
Assign a User Library, 1-146,6-9
AT&T, 1-3,E-4
ATI Wonder Card, 1-3,E-4
Auto Backup, 1-10,3-8,5-16
AutoCAD, 4-2,**4-15**

B

background printing, 1-137
BACKUP, **5-16**
Backup, 1-10,3-8
BASIC conversion programs, 8-15
Bill of Materials, 1-128,**7-4**,7-8,**7-15**
BOM1, 5-4,**7-15**

BOM2, 5-4,**7-17**
BOM3, 5-4,**7-17**
BOMx(+), **5-16**
Combine Fields, **7-18**
Header, 7-18
Length of Page, 7-19
Bipolar PROM Body Objects, F-10
BIT, 1-78
BIT 1, 3-50
BIT 2, 1-79,3-50
BIT 3, 1-78,1-79,3-51
Blank Pages, 3-10
BODY, 1-91,1-144,**3-50**,**3-130**,**6-14**
BODY field, 1-62,3-36
Body Objects, 1-58,1-62,3-35,3-42,F-1
BODY-BIT, 1-63,1-74,1-77,1-78,**3-50**
BODY-CIRCLE, **3-52**
BODY-DF PI, 1-67,1-76,1-92,3-42,3-45,**3-54**
BODY-HDR, 1-64,3-43,**3-58**
BODY-LINE, 1-63,1-74,1-79,1-83,1-109,**3-59**
BODY-N OBJ, 1-76,3-37,3-42,**3-60**
BODY-PARM, 1-85,1-110,3-42,**3-61**
BODY-PARM N, 1-66,3-62
BODY-PARM R, 1-65,3-63
BODY-PIN, 1-68,1-81,1-83,1-84,3-43,3-44,**3-63**
Hierarchy, 1-110
BODY-TEXT, 1-64,1-83,**3-66**
BOM1, 7-15
BOM2, 7-17
BOM3, 7-17
Browsing, 3-19
Files, 3-113
Libraries, 1-49,3-125

BUILD ALL, 7-2,7-9,7-12
BUS, **3-77**
Bus Connections, 3-31

C

^ ^ CONN, 3-47
C GRID, 3-6,3-7,3-16,3-67
C-Itoh, E-13
CADNETIX / SCI-CARDs, 7-39
CALAY, 7-39
CALCOMP PCI, **4-13**
Canon, E-13
CBDS Format IBM Mainframe, 7-39
CGA, 1-3,E-5
CGRID, **5-17**
CIRCLE, 3-52
Citizen, E-14
COLOR
    FOREGROUND, 2-34
    Manager, 2-33,2-36
Colors
    SCHDRAW, 5-24
COMMAND, 2-30
COMMAND PATH, 2-30
COMMAND TITLE, 2-30
COMPRESS LIBRARY 1, 6-7,6-8,**6-14**
Computer Vision, 7-39
CONFIG, 1-29,**3-67**
    PAN, 1-51
CONFIG-C GRID, 1-30,1-64,1-78,3-6,3-16,**3-67**
CONFIG-COLORS, 1-52,**3-68**
CONFIG-DIAG, 1-52,3-7,3-59,**3-69**

CONFIG-LIBR, 1-51,**3-70**
CONFIG-LIBR 1, 1-29
CONFIG-PAN, **3-70**
CONFIG-RPT, 1-51,1-117,**3-71**
CONFIG-RPT X, 1-117
CONFIG-RPT Y, 1-117
CONFIG-SAVE, **3-72**
CONFIG-V GRID, 3-16,**3-73**
CONFIG-VLABEL, **3-72**
CONFIG-WI HDR, 3-15,3-29,**3-74**
CONFIG-XY WDW, 1-52,**3-75**
CONFIG-XY Wdw, 1-29
Connections, 3-25
Connectors, 3-45
CONVERT.EXE, 8-14,B-7
COPY, 1-97,**3-93**
COPY OBJECT LIBRARY 1 TO 2, 1-142,6-7,**6-15**
CREATE A PART, 1-57
CREATE LIBRARY 2, 1-143,6-7,**6-17**
CURSOR
    SCHDRAW, 3-5
Cursor Grid, 1-52,3-6

D

dataCON Wire-Wrapping, 7-39
DBASE, 7-4,7-8
DELETE, 5-4
DELIMITED, 7-4,7-8
DeMorgan-equivalent, 3-20,3-22,3-86
DESC, 1-87,1-144,**3-131**
DESC 1, 1-144
DESC 2, 1-73

DESC1, **6-18**
DESC2, **6-19**
DESC3, **6-20**
DESC4, **6-20**
Design Rule Check, 1-128,**7-5**,7-8,**7-21**
DF PI, 1-68
DIAG, **5-19**
DIR. OF OBJECTS LIBRARY 1, 1-141,6-7,**6-21**
Discrete Component Name Objects, F-8
Discrete Name and Body Objects, 1-69
DISK BACKUPS, 1-5
DOCUMENT, 1-123
DOT, 3-29,**3-79**
DRAW, 1-15,1-46,**3-76**
DRAW-BUS, 1-46,3-31,**3-76**
DRAW-COPY, **3-78**
DRAW-DOT, 1-19,1-46,1-55,3-25,3-29,**3-78**
DRAW-ERASE, **3-80**
DRAW-GATE, 1-48,3-20,**3-81**
DRAW-LABEL, 1-20,1-46,1-52,1-93,1-119,3-26,3-29,**3-82**,5-41
DRAW-MOVE, **3-85**
DRAW-OBJECT, 1-15,1-46,1-49,1-53,1-70,1-91,3-19,3-21,3-36,**3-85**
DRAW-REPEAT, 1-46,1-51,1-117,1-118,3-31,**3-89**,5-39
DRAW-ROTATE, 1-48,3-20,**3-90**
DRAW-WIRE, 1-18,1-46,1-50,1-52,3-25,3-31,**3-91**
Drawing Title Block, 1-38,3-12
    F1, 1-39,**5-20**
    F2, 1-39,**5-20**
    F3, **5-21**
    F4, **5-21**
    F5, **5-22**
    F6, **5-22**
    F7, **5-22**

8, **5-22**
NOTES-TEXT, 3-13
Time/Date Stamp, 3-13
DRAWINGS, **2-12**
DXF format, **4-15**

E

ED SCH, **3-123**
EDIF, 7-39
EDIT, 1-21,1-96
Manager, **1-33,2-7**
SCHDRAW, **3-93**
EDIT OBJECT LIBRARY 1, 1-143,**6-22**
EDIT-COPY, 1-97,**3-93**
EDIT-ERASE, 1-97,**3-95**
EDIT-GATE, 1-48,**3-96**
EDIT-ID, 1-79,**3-97**
EDIT-IMAGE, 1-23,1-48,1-97,1-117,1-121,3-22,**3-99**
EDIT-LINE, **3-101**
EDIT-LOAD, 1-101,**3-100**
EDIT-MOVE, 1-22,1-97,**3-102**
EDIT-NEWREF, 1-101,**3-103**
EDIT-PM OBJ, 1-75,3-24,3-63,**3-104**
EDIT-REGION, 1-101,3-23,3-64,**3-105**
EDIT-REGION-LOAD, **3-100**
EDIT-REGION-SAVE, 4-25
EDIT-ROTATE, 1-48,**3-106**
EDIT-SAVE, 1-101,**3-107**
EDIT-SELECT, 1-97,3-94,3-95,3-99,**3-108**
EDIT-UPDATE, 1-80,1-86,**3-109**
EDIT-WIRE, 1-98,**3-110**
EGA, 1-3,E-5

EPROM/RAM/PAL Body Objects, F-12
Epson, E-14
Epson LQ-Series, E-15
ERASE, 1-22
ERASE OBJECT LIBRARY 1, 1-142,6-7,**6-23**
Error Messages, **F-1**
Execution Parameters
    Manager, 2-39
Expanded Memory, E-1

F

FILE, 1-27,1-93,2-29,2-33,**3-111**
    Manager, **2-8**
File Conventions, A-9
FILE-ED OBJ, 1-57,1-95,3-35,**3-111**
FILE-LOAD, 1-28,1-94,**3-112**
FILE-POP, 1-122,**3-114**
FILE-PRINT, 1-94,1-137,**3-114**,4-18
FILE-PUSH, 1-122,1-123,**3-117**
FILE-QUIT, 1-32,**3-119**
FILE-REDRAW, **1-28**,1-94,**3-119**
FILE-SAVE, 1-27,1-55,1-93,3-11,**3-120**
FILES BROWSE, **3-113**
FILTER, **2-9**,2-29,2-33
FLOATING INPUTS, 7-21
Flow Chart Objects, F-23
FutureNet, 7-39

G

GATE, 1-17,1-49,3-20,**3-81**,**3-96**
Genicom, E-16

Genoa, 1-3,E-4
Global Nets, 1-121,**7-37**
GRAPH, 1-37,5-2,5-5,5-6,**5-23**,5-25,E-2
GRAPH.LBR, F-6,F-17
GRAPHICS ADAPTER, **E-2**
Graphics Adapter, 5-2
GRAPHICS ADAPTER SELECTION, **5-25**
Ground
    Analog Ground, 3-30
    Chassis Ground, 3-30
    Digital Ground, 3-30
GWBASIC/BASICA, 8-15

H

Hardware Requirements, **F-1**
HBK, 8-3,8-4
HELP
    Manager, 1-33,**2-7**,2-32
    SCHDRAW, 1-12,1-44,3-15,3-122
Hercules Monochrome Graphics, 1-3,E-4
Hewlett-Packard Deskjet, E-16
Hewlett-Packard HP-GL, **4-12**
Hewlett-Packard LaserJet+, E-17
Hewlett-Packard PaintJet, E-17
Hewlett-Packard ThinkJet, E-17
Hidden Pins for Power and Ground, 1-84
Hierarchical Name Object, 1-111
Hierarchical Objects, 3-47
Hierarchy, 1-107,**7-31**
    Design Guidelines, 7-36
Houston Instrument DM/PL, **4-12**
HP Vectra, 1-3,E-5

HSF, 1-118,7-33

I

IBM, 1-3
IBM Color Printer, E-18
IBM Enhanced Graphics Adapter, E-5
IBM Standard Color Graphics Adapter, 1-3,E-5
ID, 1-79,**3-97**
IEEE.LBR, F-6,F-26
IEEE/IEC, F-6
IMAGE, **3-99**
INSERT, 5-4
Install
    Manual, A-3
INSTALLATION, 1-6
Intergraph, 7-39
INTERNATIONAL CHARACTERS, C-5

K

Keyboard Keys
    SCHDRAW FUNCTION KEYS, C-2
Keyboard Usage, 1-9,3-6,3-8,C-1
    International Characters, C-5
    PARTS BROWSING FUNCTION KEYS, C-4
    TEXT EDITING KEYS, C-1

L

LABEL, 3-26,**3-82**
Label Connections, 3-26

Label Swapping, **7-30**
LESSON 1, 1-7,1-12
LESSON 2, 1-7,1-33
LESSON 3, 1-7,1-41
LESSON 4, 1-7
LESSON 5, 1-7,1-91
LESSON 6, 1-7,1-107
LESSON 7, 1-7,1-125
LESSON 8, 1-8,1-137
LESSON 9, 1-8,1-139
LIB1, 6-6,**6-24**
LIB2, 6-6,**6-26**
LIBR, 3-70,**3-123**
LIBR-ED SCH, 1-80,1-89,**3-123**
LIBR-LOAD, 1-76,**3-124**
LIBR-NEW, 1-80,1-87,**3-126**
LIBR-QUIT, **3-126**
LIBR-REDRAW, **3-127**
LIBR-SAVE, 1-62,1-86,1-89,**3-128**
    Hierarchy, 1-111
    Hierarchy Name Objects, 1-113
Libraries, F-1
Library Browse, 3-88
Library Menu
    DELETE, **5-18**
    INSERT, **5-26**
LIBRx, **5-27**
Library Menus, 5-12
Library Philosophy, 1-146,6-9
LIM (LOTUS-INTEL-MICROSOFT), E-2
LINE, 1-83
LOTUS, 7-4,7-8

M

Manager, 2-1
Maximum Body Object Size, 3-48
MCGA, E-6
Merging Files, **3-101**
MODE command, 4-1,4-9
MOUSE
    Drawing Parameters Menu, **5-29**
    Middle Mouse Button, 3-8
    SCHDRAW, 1-9
MOVE, **3-85**
Multi-Color Graphics Array, E-6

N

N OBJ, 1-76
NAME, **3-130**
Name Object Editor Menu, 6-7
Name Objects, 1-58,1-87,3-35,3-38,F-1
NAME-BODY, 1-62,1-89,3-36,3-38,3-41,**3-130**,6-5
    Hierarchy, 1-113,3-49
NAME-DESC, 1-59,1-87,1-89,3-36,3-38,3-40,**3-131**
NAME-DESC 1, 1-60
    Hierarchy, 1-112
NAME-DESC 1 - 4
SCHLIB, 6-5
NAME-DESC 2, 1-60
NAME-DESC 3, 1-61
NAME-DESC 4, 1-61
Naming Conventions, 3-48
NEC MultiSync, 1-3

NEC MultiSync, E-6
Nested Objects, 1-58,1-76,3-35,3-38,3-42,F-13
Net List, 1-133,7-6,7-8,7-24,B-1
Net/Pin List Conversion, 7-8,**7-38**,B-6
NETS WITH A SINGLE PIN, 7-21
NETS WITH MULTIPLE DRIVING SOURCES, 7-21
NETS WITH NO DRIVING SOURCE, 7-21
Network Capatibility, 8-27
NEW, 6-27
    SCHDRAW, **3-126**
NEW_PAGE, 1-33,1-115,2-13,**2-15**
NEWREF, 1-101,**3-104**
Non-Component Objects, F-15
NOTES, 1-95,**3-133**
NOTES-CIRCLE, 1-24,1-95,**3-135**
NOTES-COPY, **3-133**
NOTES-ERASE, **3-134**
NOTES-LINE, 1-96,3-12,**3-136**
NOTES-MOVE, **3-136**
NOTES-TEXT, 1-26,1-96,3-13,**3-137**,5-3

O

OBJECT, **3-86**,6-8
Object Editor, 1-57,3-2,3-35,**3-111**
    Invoking, 3-35
Okidata, E-18,E-19
OMATION environment variable, A-4
Orchid Pro-Designer, 1-3,E-6

P

PageMaker, 4-24
PAN, **5-30**
Paradise, 1-3,E-6
PARM, 3-61
PARM 1, 1-72
PARM 2, 1-72
PARM 4, 1-72
PARM N, 1-66,3-62
PARM R, 3-63
PATH, **2-9**,2-29,2-33
PAUSE, 2-40
PC Paintbrush, E-20
PCAD-PDIF, 7-39
PENW, 4-5,**5-30**
Percent variables
    Manager, 2-37
Pin List, 1-133,7-6,7-8,B-3
Pin list, 7-24
PIN WITH MORE THAN ONE LABEL, 7-21
PKUNZIP.EXE, A-2
PLOTC, 4-4,**5-31**
PLOTTER, 4-4,5-2,**5-32**,E-23
Plotter Communications, 4-1
PLOTTER OPTIONS SELECTION, **5-33**
PLOTTER PROBLEM CHECKLIST, 4-11
PM OBJ, **3-104**
PostScript, **4-14**
Pound variables
    Manager, 2-37
Power and Ground, 3-43
PRINT, 1-37,**3-115**,5-2,5-5,5-6,**5-33**,E-10

PRINTER DRIVER SELECTION, 5-34
Printer Drivers, E-10
Printronix, E-21
PROMPT, 5-4,**5-35**
Proprinter, E-21
PSPCONV.EXE, 8-22
PSpice, 7-39
    Digital/Analog, 8-22
PUSH, **3-118**
PXLEN, 4-4,4-6,**5-36**
PXOFF, 4-5,**5-37**
PYLEN, 4-4,4-6,**5-36**
PYOFF, 4-5,**5-37**

Q

QUIT
    Manager, **2-10**
    SCHCFG, **5-38**

R

Racal Redac, 7-39
Ram Drive, 7-40
REDRAW, 3-80,**3-120**
REFD, 1-65
Reference Designators, 3-22,3-87
REGION, 1-101
REGION-COPY, 1-101,1-102,3-94
REGION-ERASE, 1-101
REGION-LOAD, 1-103
REGION-MOVE, 1-102
REGION-SAVE, 1-103

REPEAT, **3-89**
Reports, 1-127,**7-7**,7-8,**7-27**
ROTATE, 1-17,1-49,3-20,**3-90**,**3-107**,F-7
ROTATED Nested Objects, F-51
Routing, 1-127,**7-2**,**7-13**
RPTX, **5-38**
RPTY, **5-38**

S

S_ROUTER, 2-26
SAVE
    Manager, **2-11**
SCHBACK, 2-16,**8-1**
    Invoking, 8-4
    User Messages, 8-6
SCHCFG, 1-36,2-16,F-1,A-6
EXIT, **5-19**
LOAD, 5-12,**5-27**
SAVE, 5-12,**5-39**
SCHDRAW, 2-17,3-1
    Invoking, 3-2
    Screen Format, 3-3
SCHECO, 2-23,**8-7**
    Invoking, 8-7
SCHEMA-PCB Pin List, 7-9
SCHEMA.BAT, 2-2
SCHEMA.CFG, 1-37,5-1,5-2,A-6
SCHEMA.MGR, 2-2,2-28
Schematic Editor, 1-44,3-1
SCHLIB, 1-139,2-18,F-1
    BODY, 6-8
    DESC1, 6-8

Invoking, 6-2
LIB1, 1-140
LIB2, 1-141
NEW, 1-144,6-8,**6-27**
OBJECT, 6-8,**6-27**
PRINT LIBRARIES, 6-21
QUIT, **6-28**
SAVE, 6-8,**6-29**
SCHLOCAL, 2-18,**8-9**
Invoking, 8-10
SCHPAGE, 2-19,4-17,**8-11**
Invoking, 8-12
SCHPCB, 2-24
SCHPLOT, 1-137,2-20,3-32,**4-1**
Invoking, 4-3
SCHPOST, 1-125,2-21,**F-1**
Invoking, 7-7
SCHPRINT, 1-137,2-22,3-32,**4-18**
Invoking, 4-19
SCHTEMP environment variable, 7-40
SDF, 7-4,7-9
SEARCH, 1-30,1-103,3-22,**3-141**
SEARCH-LABEL, 1-31,1-105,**3-141**
SEARCH-NAME, 1-104,**3-142**
SEARCH-REFD, 1-105,**3-144**
SEARCH-TEXT, 1-105,**3-145**
SEARCH-X,Y, 1-103,**3-146**
SEDIT, 2-24
SELECT, **3-108**
Sigma, 1-4,E-7
Signal Connections, 3-25
SLIB, 2-25
SOTA, 1-4

SREAD, 2-25
Star, E-21
STATUS, 2-28
STATUS LIBRARY 1 OR 2, 1-142,6-7,**6-30**
STB, E-7
SWAP LIBRARIES 1 AND 2, 1-142,6-6,**6-31**
SWRITE, 2-26

T

Tecmar, 1-4,E-7
Texas Instruments 850, E-22
TIFF, **4-24**,E-22
Timing Diagram Objects, F-24
TITLE, 2-29,2-33
Title Block Variables, 3-13,**3-138**
Toshiba, E-7,E-22
Toshiba T3100, 1-4
Trident Impact, 1-4,E-5,E-8

U

UNDO, 1-9,1-97,1-102,3-7,3-80,3-95,3-135,**3-147**,E-9
UPDATE, **3-109**

V

V GRID, 1-51,3-16,3-73
VEGA Video7, 1-4
Ventura, 4-24
VGA, E-8
VGRID, **5-40**

Video 7, E-8
Video Graphics Array, 1-3,E-8
VIEW, **2-11**
Virtual Memory, 7-40
Visionics EE Designer, 7-39
Visual Grid, 1-51
VLABEL, **5-41**

W

was/is, 8-2
WI HDR, 3-15,3-26,3-29,**3-74**
WIHDR, **5-41**
WIRE, 3-31,**3-91**
Wire List, 1-133,**7-6**,7-9,7-24
Word, 4-24
WordPerfect, 4-24
Wyse, 1-4,E-8

X

X,Y, 1-103
XMOUSE, 2-33
XY, **5-42**

Y

YMOUSE, 2-33

Z

ZOOM, 1-41,1-45,**3-148**,5-42
SCHCFG, **5-42**
SCHDRAW, 1-13
Zoom IN, 1-45
Zoom OUT, 1-45
ZOOM-VGA, 1-45

<Left Blank>

WARRANTY/END-USER SUPPORT

Warranty/end-user support is available from Omation. Please contact Omation Customer Service at (214) 231-5167 for the following information:

* Clarification of documentation
* Solutions to known problems
* How to replace defective disks or lock devices.

The form on the following page has been included to aid problem reporting. If you encounter a problem, please complete the form (if necessary, the form may be copied) and mail it to the following address:

Omation, Inc.
Attn: Customer Service
801 Presidential
Richardson, Texas 75081

OMATION SOFTWARE/DOCUMENTATION PROBLEM CARD

PLEASE MAIL THIS CARD TO OMATION TO REPORT SOFTWARE PROBLEMS, DOCUMENTATION ERRORS, SUGGESTIONS, ETC.

Name _____

Street _____

City _____ State _____ Zip _____

Phone ( ) _____ Date Problem Found _____

Software Problem ☐ Documentation Error ☐

Library Error ☐ Suggestion ☐ Other ☐

Lock Device Serial Number/Software Version Number

_____

Hardware Used: PC Manufacturer _____

Amount of RAM _____ Mouse _____

Graphics Adapter Type _____

DOS Version _____ Floppy Disk ☐

Hard Disk ☐

Please describe the problem below. If a software problem, please describe how and when the problem occurs. For documentation errors, please note the applicable page and list the error.

SCHEMA *PowerTools*

Reference Manual

Copyright OMATION, Inc. 1992

OMATION, Inc.
801 Presidential
Richardson, TX 75081

SUPPORT:	1-800-562-9085
SALES:	1-800-553-9119
FAX:	1-214-783-9072
BBS:	1-214-231-1350

Table of Contents

Introduction 1
Installation 3
I. ANNOTATE - Project Annotator 4
INVOKING ANNOTATE 4
RUNNING ANNOTATE 4
Main Menu ANNOTATE Commands 6
II. SEARCH - Search and Replace 8
INVOKING SEARCH 8
RUNNING SEARCH 8
Main Menu SEARCH Commands 10
III. LIBMANGR 14
Library Information Manager 14
Main Menu LIBMANGR Commands 16
Menu for GENERATE OBJECT PAGE LISTING 17
Main Menu GENERATE NAME DIRECTORY LISTING 20
IV. MAKEPAGE - Page Maker 23
INVOKING MAKEPAGE 23
RUNNING MAKEPAGE 23
Main Menu MAKEPAGE Commands 25
V. SHOWNET - Net Display Program 27
INVOKING SHOWNET 27
RUNNING SHOWNET 27
Main Menu Net Display Program 29
VI. SORTPAGE - Page Sort Program 30
INVOKING SORTPAGE 30
RUNNING SORTPAGE 31
VII. HIERPRNT - Hierarchical Print 32
INVOKING HIERPRNT 32
RUNNING HIERPRNT 32
VIII. HIERLOCL - Hierarchical Local 33
INVOKING HIERLOCL 33
RUNNING HIERLOCL 33

Table of Contents

IX. HEXEDIT - Hexedecimal Editor 34
INVOKING HEXEDIT 34
RUNNING HEXEDIT 34
Main Menu HEXEDIT Commands 36
X. Command Macros 39
INVOKING Command Macros 39
RUNNING Command Macros 39
XI. Printer Driver Source 41
Creating the Drivers using the MAKE File 41

Introduction

SCHEMA has been one of the standards in schematic capture since 1985. Thousands of engineers have made it their personal as well as professional choice for PC based design of schematics.

SCHEMA ***PowerTools*** has been developed for that designer who is a consistent user of SCHEMA. ***PowerTools*** gives the user a set of enhancement programs that will aide in the productivity of designing schematics with SCHEMA.

I. ANNOTATE - Project annotator. Updates the Reference designators automatically and assigns the reference designator prefices of the user's choice. Packs available gate sections before continuing to the next Reference designator.

II. SEARCH - Performs a Search or Search and Replace for all the pages of a schematic project.

III LIBMANGR - Creates a graphical part directory and a Name Object hierarchical tree of Body and Nested Objects.

IV. MAKEPAGE - Allows the user to create a page. The page can automatically create a border and paste in a title block region.

V. SHOWNET - Displays the pins and sheet location of a selected net of a project.

VI. SORTPAGE - Sorts the order of objects within a schematic page. This yields more efficient plotting when used with SCHPLOT.

VII. HIERPRNT - Prints all top level and hierarchical pages of a design.

VIII. HIERLOCL - Localizes the libraries of all top level and hierarchical pages of a design.

IX. HEXEDIT - Hexidecimal editor that allows the user to modify a file at the HEX/ASCII level of a fixed length file.

X. Command Macros - Utilizes ANSI.SYS to set up a series of macros that let the user jump to commands in SCHDRAW quickly.

XI. Printer Driver Source - All the source code to modify or create a printer driver of your own.

Installation

First make a backup of the distribution diskette. Do this using DOS DISKCOPY. Example:

DISKCOPY A: A: <Enter>

Next place the diskette into a floppy drive A: and type:

INSTALL <Enter>.

INSTALL will find where SCHEMA is currently installed and will display this path. During installation the SCHEMA.MGR manager file will be backed up to SCHEMA.BAK. The default manager column for the ***PowerTools*** will be installed to column 6. If you have commands already in this column don't worry, they are still in the manger file. The manager displays the last command described for a cell and since the ***PowerTools*** manager was appended to the end of the file, those are the commands displayed.

I. ANNOTATE - Project Annotator

The ANNOTATE program will auto-assign reference designators and fill available gate sections for components. The program will change all designators for a project in a sequential order from the first to the last page of a design. ANNOTATE references a reference designator preference file that gives a list of desired prefices. This preference file is an ASCII text file and can be modified to suit the user's own specifications. ANNOTATE will search through all the libraries listed in the loaded config file to count the number of gate sections found for a particular part. Section "-1A" will be searched followed by "-1B", "-1C", etc. This total number will be used to pack the gates for a given Reference designator before a new designator is selected for the part. *Note:* The user's own library parts should follow the SCHEMA library convention for creation of parts and gate sections.

INVOKING ANNOTATE

ANNOTATE can be invoked from the Manager or DOS prompt. The format is:

ANNOTATE DRAWING <Enter>

where DRAWING is the project name with no extension.

RUNNING ANNOTATE

When ANNOTATE loads, the screen is cleared and the ANNOTATE Main Menu, shown in figure 1, is displayed.

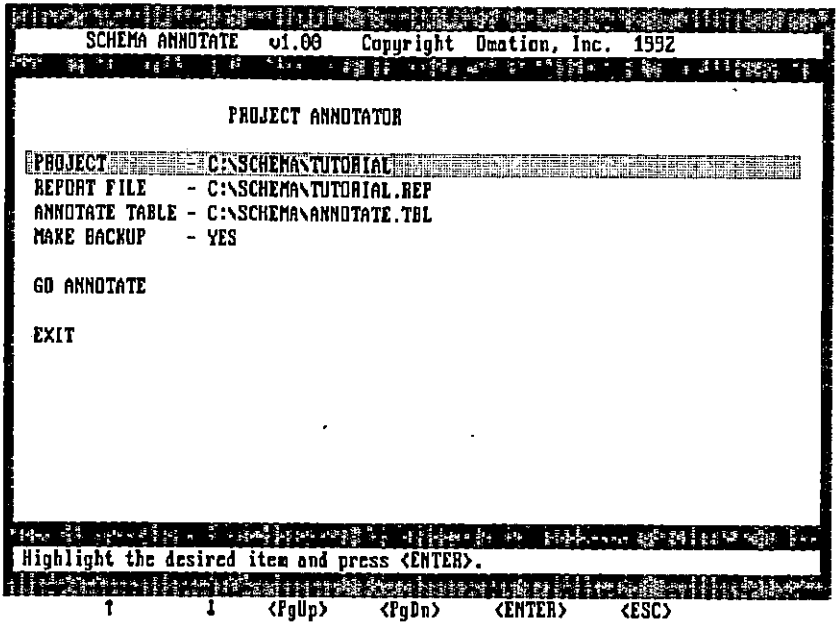


Figure 1. ANNOTATE Main Menu

The command columns list available ANNOTATE commands. To select a command, highlight the command using the mouse or the keyboard arrow keys and press **<Enter>**.

Check that the Project, Report, and the Annotate Preference Table filenames are correct. Select whether you wish to backup the schematic pages. When ready, select GO ANNOTATE.

Main Menu ANNOTATE Commands

PROJECT - Describes the path and filename of the project that the user wishes to annotate with new reference designators. Select the command and the text line below will open with the project path and filename displayed for editting. Make the appropriate changes and press **<Enter>**.

REPORT FILE - Where the ouput from the annotation process will be stored. The report will contain three sections.

The first section will be displayed as WAS/IS and Location.
Example:

74LS08,.AAA01 , 74LS08,U1 - AT : 33.00, 7.08

The second section will describe the part name, the last reference designator used, the total number of gate sections found (1A, 1B, 1C, etc), and the last gate used. Example:

74LS08 U11 4 C

The third and last section will describe all of the prefices found in the Annotate Preference Table and what the last numerical use of that prefix. Example:

U 11
R 21
C 13

ANNOTATE TABLE - Represents the filename of the Annotate Preference Table. This path and file are defaulted to the file that was supplied with this program. If you want to modify this path, position the cursor over the command and press **<Enter>**. The text line will appear for editting.

The format of the table is a part followed immediately by a comma and the reference designator prefix that can be up to four characters long. The wildcard character, '*' can be used with the part name to reference a family of parts. Example:

DEFAULT,U
DRES,R
CAP*,C

MAKE BACKUP - The command toggles between "YES" and "NO". The selection "YES" will backup the schematic pages. This is the default mode.

GO ANNOTATE - This command initiates the annotation process.

Note: ANNOTATE will not modify the reference designators for any connector parts that begin with "^^", ".C"(circles/arcs), or DOT.

EXIT - Exit the ANNOTATE program.

II. SEARCH - Search and Replace

The SEARCH program will globally Search or Search and Replace: Part names, Reference Designators, Parms 1-4, Labels, or Text for all the pages of a project. A Replace file can be used if there are multiple runs of searching and replacing to be performed on the schematic.

INVOKING SEARCH

SEARCH can be invoked from the Manager or DOS prompt. The format is:

SEARCH DRAWING <Enter>

where DRAWING is the project name with no extension.

RUNNING SEARCH

When SEARCH loads, the screen is cleared and the SEARCH Main Menu, shown in figure 2, is displayed.

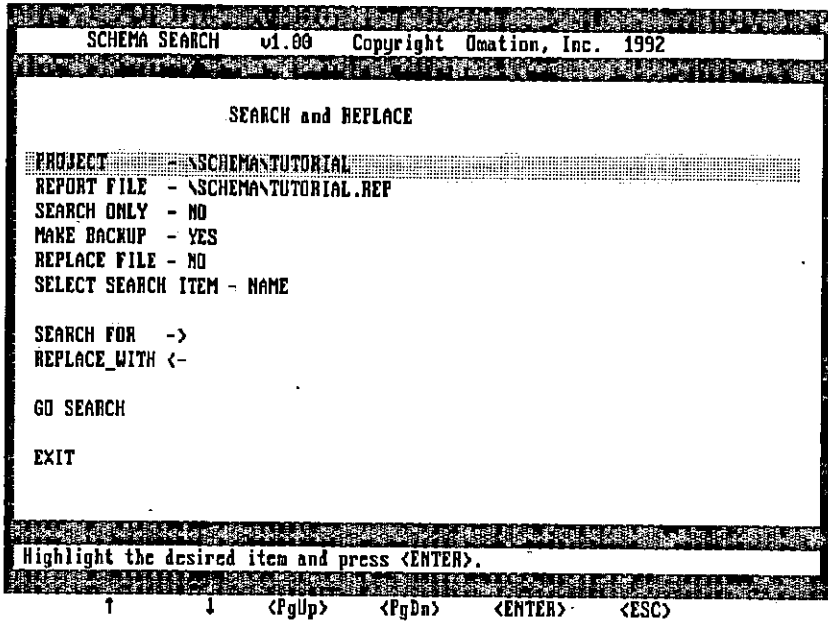


Figure 2. SEARCH Main Menu

The command columns list available SEARCH commands. To select a command, highlight the command using the mouse or the keyboard arrow keys and press **<Enter>**.

Check that the Project, Report, and the SEARCH FOR and REPLACE WITH command lines are correct. Select whether you wish to backup the schematic pages, search only, or use a replace file. Use SELECT SEARCH ITEM, to indicate which SCHEMA parameter to use. When ready, select GO SEARCH.

Main Menu SEARCH Commands

PROJECT - Describes the path and filename of the project that the user wishes to annotate with new reference designators. Select the command and the text line below will open with the project path and filename displayed for editting. Make the appropriate changes and press **<Enter>**.

REPORT FILE - Where the ouput of the SEARCH process will be stored. The format of the report describes the search item, the page processed, where found, and replaced with what. Example:

SEARCHING FOR : 74LS08
PROCESSING - PROJECTX.P01
FOUND AT : X - 17.00 Y - 49.08
REPLACED WITH: 74ALS08

SEARCH ONLY - Toggles between "YES" and "NO". "NO" is default. If activated, the REPLACE WITH <- command is disabled. SEARCH will only look for an item and will not attempt to replace it with anything.

MAKE BACKUP - The command toggles between "YES" and "NO". The selection "YES" will backup the schematic pages. "YES" is the default mode.

REPLACE FILE - This command is default set to "NO". If selected, it creates a file path using the project name and the extension, ".RPL". The path can be altered if it is not correct. If activated, the SEARCH FOR -> and the REPLACE WITH <- commands are disabled. SEARCH will read the replace file sequentially to update the pages.

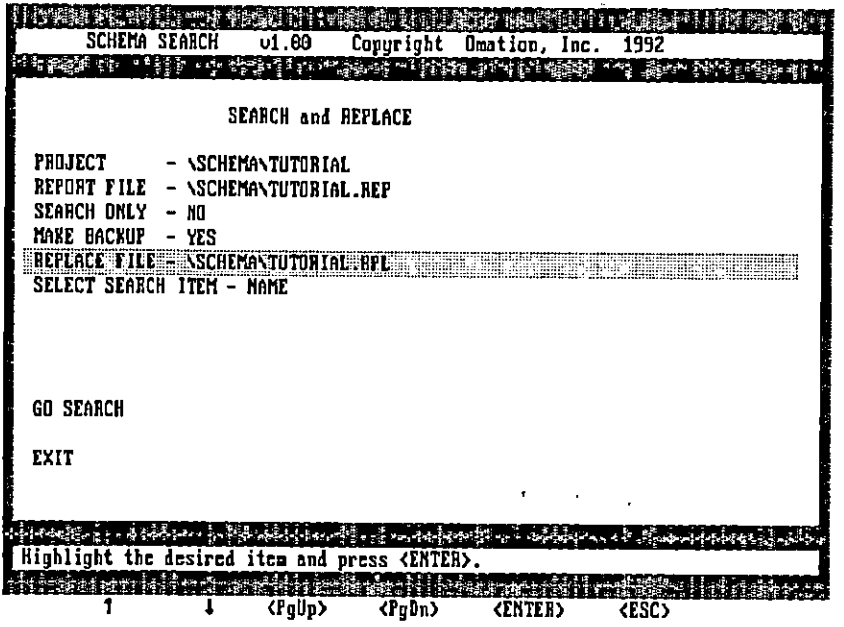


Figure 3. Replace File for SEARCH

The format of the replace file is as follows:

X | SEARCH String | REPLACE String

where X represents the Search Item. The delimiter is the vertical bar '|'.

N - Name
R - Reference Designator
L - Label
T - Text
1 - Parm1
2 - Parm2
3 - Parm3
4 - Parm4

Examples:

N | 74LS08 | 74ALS08
R | U20 | R20
L | CLK | CLOCK
T | RESULTS TESTA | RESULTS TESTB
1 | 10K | 100K

Note: If Search Only is activated while using the replace file, no items will be replaced only a report as to where these items were found.

SELECT SEARCH ITEM - Clears the main menu of SEARCH and displays the Item menu. Choose which of the eight items that are to be searched. The choices are: **NAME, REFD, LABEL, TEXT, PARM1, PARM2, PARM3**, and **PARM4**. **NAME** is default.

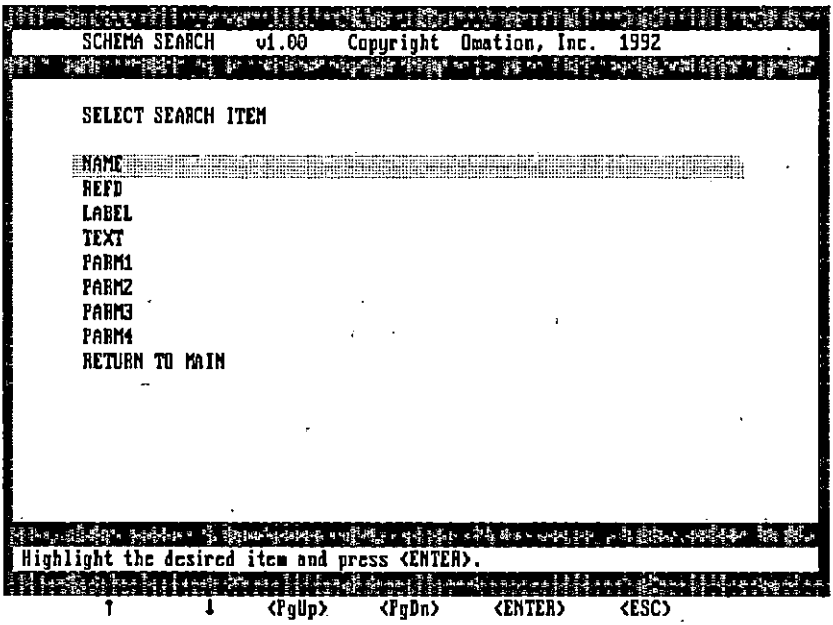


Figure 4. SEARCH Select Item Menu

SEARCH FOR -> Enter the string to search for in the SCHEMA pages. The wildcard character, '*' can be used only if SEARCH ONLY is set to "YES".

REPLACE WITH <- Enter the replace string with which to replace the found entries.

GO SEARCH - Activates the search process. A chance will be given to the user at the beginning whether or not SEARCH should prompt before replacing items.

EXIT - Exit the SEARCH program.

III. LIBMANGR

Library Information Manager

The LIBMANGR program has two sections.

One section creates for a selected library a series of pages using a template page. The graphical body object parts are placed in alpha numeric order on the pages starting in the upper left hand corner at 4,4 (64,64 pixels). Each part has its name bounded by a thin line in the upper left. These series of pages can be viewed or printed and provide a graphical directory of the library objects.

The other section creates for a selected library a document file that lists a Name Object, its descriptions, Body field, the extensions found for that Body field, and the Nested Objects used by all of the Body Objects. This output provides a hierarchical tree for each Non-graphical part.

These two outputs together create a useful library management tool.

INVOKING LIBMANGR

LIBMANGR can be invoked from the Manager or DOS prompt. The format is:

LIBMANGR LIBRARY <Enter>

where LIBRARY is the library from which the user wants a directory.

RUNNING LIBMANGR

When LIBMANGR loads, the screen is cleared and the LIBMANGR Main Menu, shown in figure 5, is displayed.

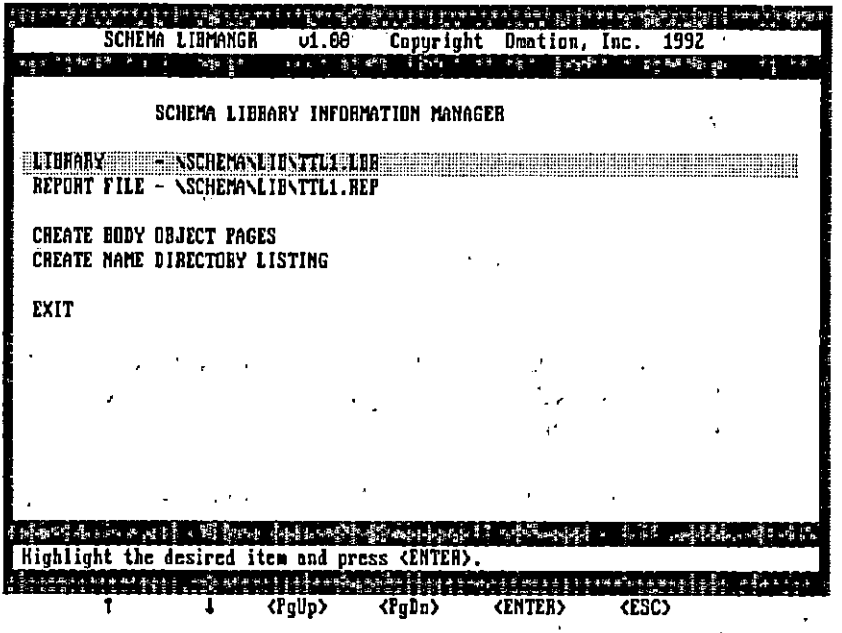


Figure 5. LIBMANGR Main Menu

The command columns list available LIBMANGR commands. To select a command, highlight the command using the mouse or the keyboard arrow keys and press **<Enter>**.

Check that the Library, and Report filenames are correct. When ready, select either CREATE BODY OBJECT PAGES or CREATE NAME DIRECTORY LISTING.

Main Menu LIBMANGR Commands

LIBRARY - When selected the library path and filename are given at the text line below. Make any changes as necessary. The wildcards '*' and '?' can be used to invoke a list of libraries from which to choose.

REPORT FILE - Where the error messages for the LIBMANGR process will be stored.

CREATE BODY OBJECT PAGES - Invokes the GENERATE OBJECT PAGE LISTING menu. This portion of the program will create the graphical listing of objects.

CREATE NAME DIRECTORY LISTING - Invokes the GENERATE NAME DIRECTORY LISTING menu. This portion of the program will create the Name Object hierarchical directory listing.

EXIT - Exits the LIBMANGR program

Menu for GENERATE OBJECT PAGE LISTING

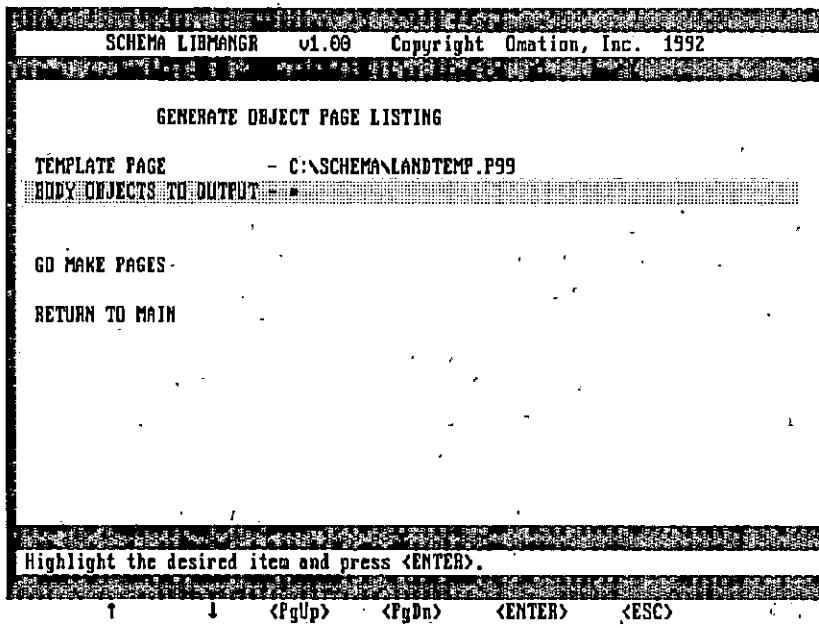


Figure 6. Main Menu Generate Object Page Listing

TEMPLATE PAGE - Shows the template page to create the object pages. The default is LANDTEMP.P99. This page is provided with the program and is a landscape style page that will fit a standard 8.5 x 11 inch paper from a HP LaserJet in the default print mode (150 dpi). Another template page supplied with the software is PORTTEMP.P99. This page is in a portrait style. The user can create and use any template page. Select from a list of pages by using the wilcard characters '*' and '?'. Example: \SCHEMA*.P99.

BODY OBJECTS TO OUTPUT - Indicates what parts you wish to be placed on the graphical directory pages created. The default is '*', which is all graphical objects. Meaning any object that DOES NOT have a Description 1 - 4 or a Body Field. A grouping of parts can be requested by typing in a prefix. Example:

T08-*

This will ouput all the Body objects that begin with a "T08-".

GO MAKE PAGES - Activates the Create Object Page process. The page name will be displayed as they are created on the screen. If the library requires more pages than ".P98", the the prefix letter will switch to 'Q', 'R', 'X', and then 'Y'. A config file for the library will be created and placed in the same directory as the pages. The **LIBR1** parameter will have the library file and path described. The Drawing Title Block Parameters will be added. The **F1** parameter will have the library name. The **F5** parameter will have the maximum number of pages.

Note: If a drawing or config file currently exists, the user will be prompted to whether or not the file should be overwritten.

RETURN TO MAIN - Returns the the main LIBMANGR program.

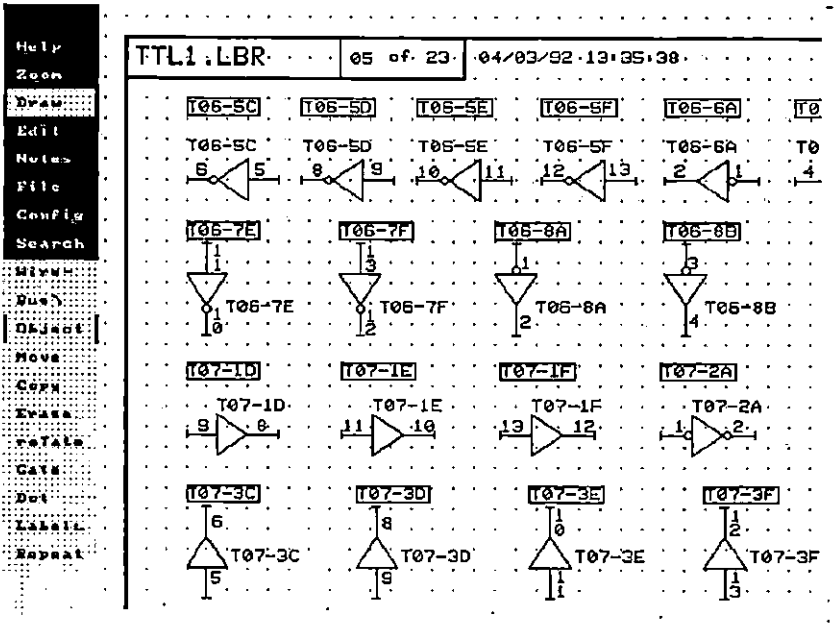


Figure 7. Sample Object Page Listing

Main Menu GENERATE NAME DIRECTORY LISTING

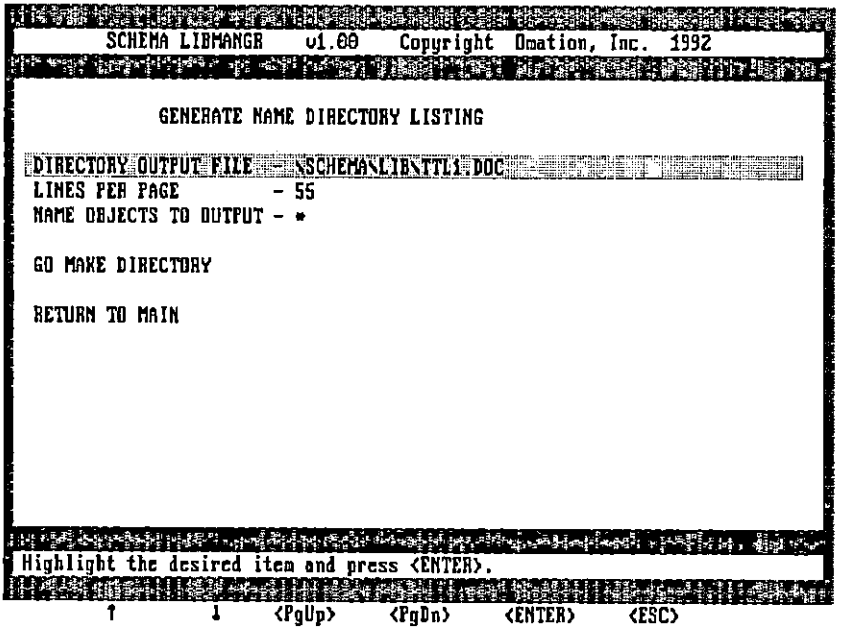


Figure 8. Main Menu Generate Name Directory Listing

DIRECTORY OUTPUT FILE - This file is the library path and file prefix with a ".DOC" extension. The file will list each of the Name Objects and their respective Body Objects and Nested Objects. If you wish the output to go directly to the printer, type "PRN" as the Directory Output File.

LINES PER PAGE - This is the number of lines to output to a page before a form feed is requested. "55" is the default.

NAME OBJECTS TO OUTPUT - Indicates what Name Object parts you wish to be sent to the document pages created. The default is '*', which is all Name objects. Meaning any object that DOES have a Description 1 - 4 or a Body Field. A grouping of parts can be requested by type in a prefix. Example:

74LS*

This will ouput all the Name objects that begin with a "74LS".

GO MAKE DIRECTORY - Activates the Name Directory Listing process. The Name object names will be displayed as they are processed. The output document will have a header line with a File/Time/Date Stamp.

RETURN TO MAIN - Returns to the Main LIBMANGR program.

```
LIBR: TTL1.LBR      DATE: Fri Apr 03 13:41:26 1992      PAGE: 1

74LS04
DESC1  - 300-0004-001
DESC2  - 74LS04 - LS TTL HEX INVERTER
BODY   - T04-1A, 1B, 1C, 1D, 1E, 1F, 2A, 2B, 2C, 2D, 2E, 2F, 3A, 3B
             3C, 3D, 3E, 3F, 4A, 4B, 4C, 4D, 4E, 4F, 5A, 5B, 5C, 5D, 5E
             5F, 6A, 6B, 6C, 6D, 6E, 6F, 7A, 7B, 7C, 7D, 7E, 7F, 8A, 8B
             8C, 8D, 8E, 8F
NESTED - OINV, OINVD, OINV090, MBUF090, OINV180, MBUF180,
OINV270, MBUF270

74LS05
DESC1  - 300-0005-001
DESC2  - 74LS05 - LS TTL HEX INVERTER WITH OC OUTPUT
BODY   - T06-1A, 1B, 1C, 1D, 1E, 1F, 2A, 2B, 2C, 2D, 2E, 2F, 3A, 3B
             3C, 3D, 3E, 3F, 4A, 4B, 4C, 4D, 4E, 4F, 5A, 5B, 5C, 5D, 5E
             5F, 6A, 6B, 6C, 6D, 6E, 6F, 7A, 7B, 7C, 7D, 7E, 7F, 8A, 8B
             8C, 8D, 8E, 8F
NESTED - OINV, OINVD, OINV090, MBUF090, OINV180, MBUF180,
OINV270, MBUF270

74LS08
DESC1  - 300-0008-001
DESC2  - 74LS08 - LS TTL QUAD 2 INPUT AND GATE
BODY   - T08-1A, 1B, 1C, 1D, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, 4A, 4B
           4C, 4D, 5A, 5B, 5C, 5D, 6A, 6B, 6C, 6D, 7A, 7B, 7C, 7D,8A
         8B, 8C, 8D
NESTED - OAND, OANDD, OAND090, ONOR090, OAND180, ONOR180,
OAND270, ONOR270
```

Figure 9. Sample Name Object Directory Listing

IV. MAKEPAGE - Page Maker

The MAKEPAGE program will allow the user to create a SCHEMA page for use with SCHDRAW. The page X,Y values must be multiples of 8. There is a selection for a border, which will draw a thick line border 32 pixels or 2 grid units from the edge of each side. The Number of divisions are defaulted to what most users create. 8 divisions on the top and bottom and a 'D' size with 4 divisions on the left and right The border is divided by markers on all sides depending on these values. A title block region can be pasted onto the lower right hand corner, if requested. This region can be modified to fit a users particular title block format.

INVOKING MAKEPAGE

MAKEPAGE DRAWING <Enter>

where DRAWING is the project name with no extension.

RUNNING MAKEPAGE

When MAKEPAGE loads, the screen is cleared and the MAKEPAGE Main Menu, shown in figure 10, is displayed.

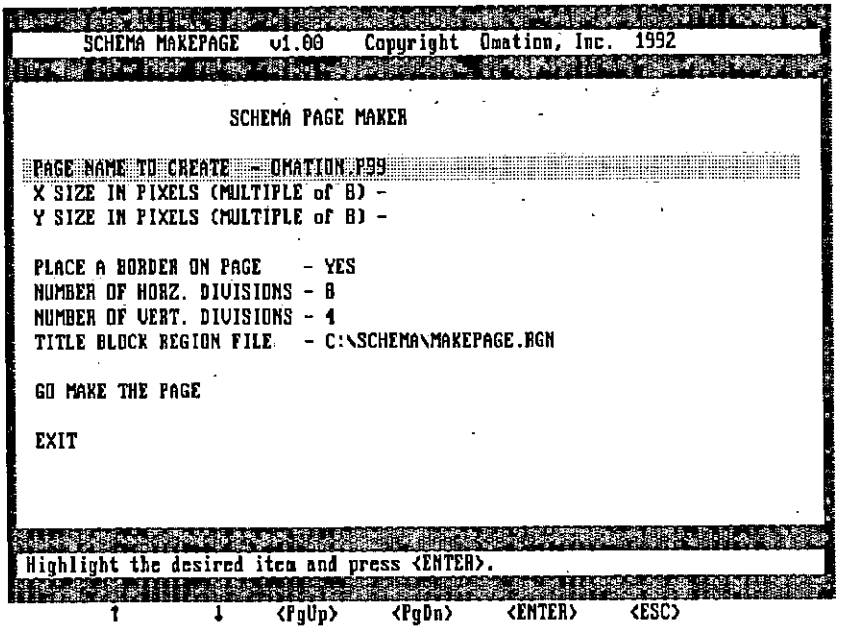

SCHEMA MAKEPAGE v1.00 Copyright Omation, Inc. 1992
SCHEMA PAGE MAKER
PAGE NAME TO CREATE - OMATION.P99
X SIZE IN PIXELS (MULTIPLE of 8) -
Y SIZE IN PIXELS (MULTIPLE of 8) -
PLACE A BORDER ON PAGE - YES
NUMBER OF HORZ. DIVISIONS - 8
NUMBER OF VERT. DIVISIONS - 4
TITLE BLOCK REGION FILE - C:\SCHEMA\MAKEPAGE.RGN
GO MAKE THE PAGE
EXIT
Highlight the desired item and press <ENTER>.
↑ ↓ <PgUp> <PgDn> <ENTER> <ESC>

Figure 1. MAKEPAGE Main Menu

The command columns list available MAKEPAGE commands. To select a command, highlight the command using the mouse or the keyboard arrow keys and press **<Enter>**.

After insuring that the Page Name is correct, the X,Y size is a multiple of 8. Then decide if you wish to have a border. If "YES", then select the number of horizontal and vertical divider markers. Decide whether to paste a title block in the lower right hand corner. When ready, select GO MAKE THE PAGE.

Main Menu MAKEPAGE Commands

PAGE NAME TO CREATE - Type in a filename and use the SCHEMA Page convention of using ".P99" as the extension.

X SIZE IN PIXELS - Type the X size for the page. Be sure to give a value that is a multiple of **8**.

Y SIZE IN PIXELS - Type the Y size for the page. Be sure to give a value that is a multiple of **8**.

PLACE A BORDER ON THE PAGE - Toggles between "**YES**" and "NO". Default is "**YES**". If de-activated the following commands dissappear: NUMBER OF HORZ. DIVISIONS, NUMBER OF VERT. DIVISIONS, and TITLE BLOCK REGION FILE

NUMBER OF HORZ. DIVISIONS - Number of divisions across the top and bottom of the page. These divions are numerically marked. The maximum is '**26**'. Default is '**8**'.

NUMBER OF VERT. DIVISIONS - Number of divisions along the left and right of the page. These divisions are alphabetically marked. The maximum is '**26**'. Default is '**4**'.

TITLE BLOCK REGION FILE - The Title block region file is defaulted to the file MAKEPAGE.RGN where the SCHEMA software should be loaded. If this is not the proper path, modify it. The region itself can be modified to reflect the user's corporate requirements. Use SCHDRAW and FILE-LOAD. Type the full file name including the ".RGN" extension. ***NOTE:*** There is a one grid unit boundary. This is necessary to paste this region exactly on the border in the lower right hand corner of the page.

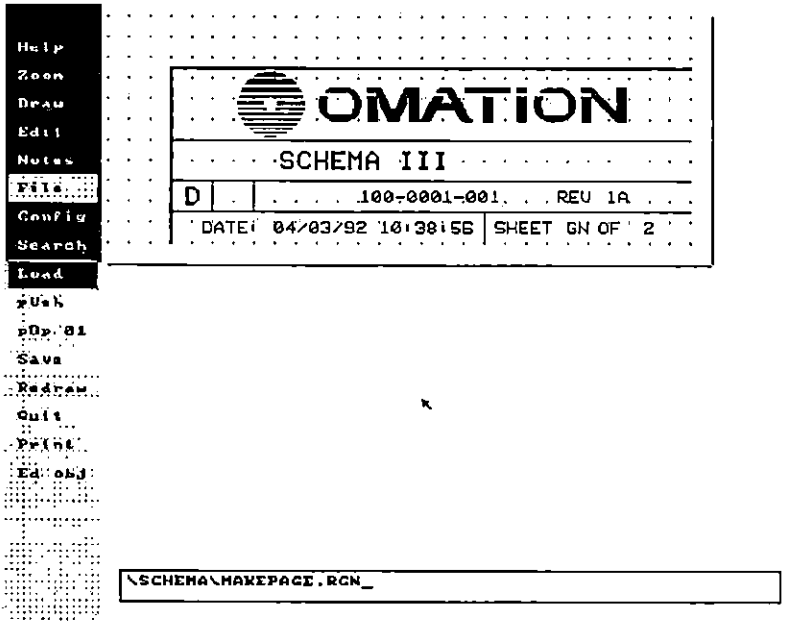


Figure 11. The MAKEPAGE.RGN Title Block Region

GO MAKE THE PAGE - Creates the page per the options selected. If the values for X and Y size or number of divisions are incorrect, the operation will be aborted.

EXIT - Exit the MAKEPAGE program.

V. SHOWNET - Net Display Program

The SHOWNET program will read a Net/Pinlist file, ".NPL" and display the pins attached to that net and the schematic page where each pin is found.

INVOKING SHOWNET

SHOWNET can be invoked from the Manager or DOS prompt. The format is:

SHOWNET DRAWING <Enter>

where DRAWING is the project name with no extension.

RUNNING SHOWNET

When SHOWNET loads, the screen is cleared and the SHOWNET Main Menu, shown in figure 12, is displayed.

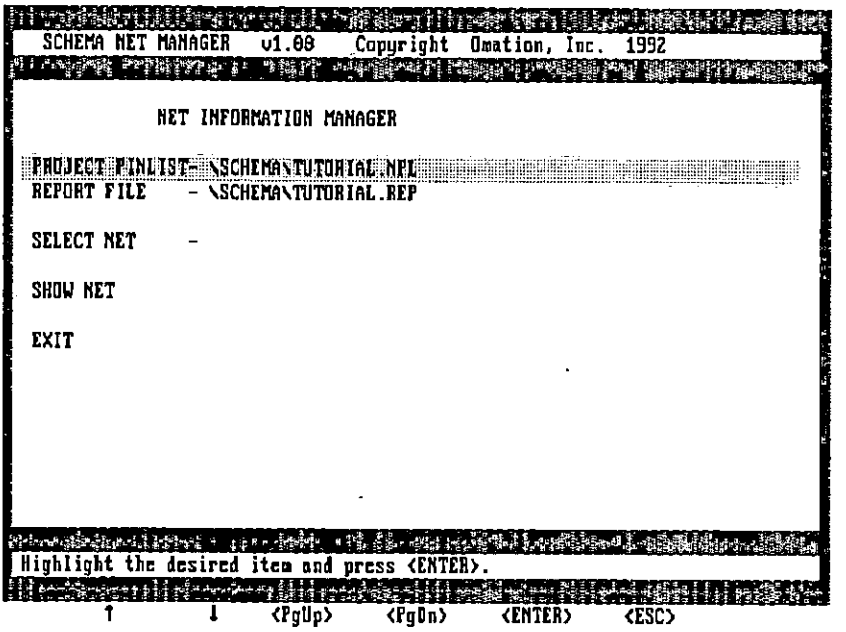


Figure 12. SHOWNET Main Menu

The command columns list available SHOWNET commands. To select a command, highlight the command using the mouse or the keyboard arrow keys and press **<Enter>**.

After ensuring that the Project file name, Report filename, and the net selected are correct. When ready, select SHOW NET.

Main Menu Net Display Program

PROJECT PINLIST - Displays the project pinlist that is to be read as input to display a net.

REPORT FILE - This is the file name where the output from the show net requested is copied. The path and file are taken from the project path and filename with a ".REP" extension.

SELECT NET - Type in the net that is to be displayed. Remember that net names are case sensitive.

SHOW NET - When ready to view the net, select this command.

EXIT - Exit the SHOWNET program.

VI. SORTPAGE - Page Sort Program

The SORTPAGE program will sort the order of the page entries of a schematic page. SCHPLOT plots page entries in the order they are found in the file. After several editting sessions on the page the order of entries may jump from one side to another and seem erratic to the user who is plotting the page. Older plotters may lose their registration as they are sent back and forth across the page. SORTPAGE will start in the upper left hand corner of the board, which is the page origin. Then SORTPAGE will search through all the page entries to find the entry that is closest to that part. Next, it will record the ending point of that entry and find the next entry that is closest to that point. SORTPAGE will continue until all the entries are sorted. The algorithm will follow contiguous lines or wires and may end up at another section of the page. The end result of the output is a smoother flow for the plotter and a more logical drawing order that will yield better plotting results. *Note:* The sorting may also be helpful in use with the ANNOTATE program.

INVOKING SORTPAGE

SORTPAGE can be invoked from the Manager or DOS prompt. The format is:

SORTPAGE DRAWING <Enter>

where DRAWING is the project name with no extension, which will process all the pages that are sequentially ordered, or a particular drawing page. Example: DRAWING.P02.

RUNNING SORTPAGE

When SORTPAGE loads, the screen is cleared, and each page name is displayed. A series of dots displayed afterwards indicates that the program is processing that particular page. Each page is backed up before the new sorted page is written to disk.

VII. HIERPRNT - Hierarchical Print

The HIERPRNT program will call the SCHPRINT program and print all the top level pages of the hierarchical design. Then HIERPRNT will read the ".HSF file and print all the hierarchical pages found within the file. HIERPRNT will pass all the same switch settings to SCHPRINT.

INVOKING HIERPRNT

HIERPRNT can be invoked from the Manager or DOS prompt. The format is:

HIERPRNT DRAWING <Enter>

where DRAWING is the project name with no extension.

RUNNING HIERPRNT

Be sure that the project pages have been processed using, SCHPOST. This is necessary because SCHPOST creates a ".HSF" file that describes all the hierarchical objects used. When HIERPRNT is run the user will see SCHPRINT called for each set of pages.

All switch settings are passed through to the call to SCHPRINT. See SCHPRINT in Chapter 4 of the SCHEMA User's Manual for a list of switches.

VIII. HIERLOCL - Hierarchical Local

The HIERLOCL program will call the SCHLOCAL program and localize the libraries all the top level pages of the hierarchical design. Then HIERLOCL will read the ".HSF file and will localize all the libraries for all the hierarchical pages found within the file.

INVOKING HIERLOCL

HIERLOCL can be invoked from the Manager or DOS prompt. The format is:

HIERLOCL DRAWING <Enter>

where DRAWING is the project name with no extension.

RUNNING HIERLOCL

Be sure that the project pages have been processed using, SCHPOST. This is necessary because SCHPOST creates a ".HSF" file that describes all the hierarchical objects used. When HIERLOCL is run the user will see SCHLOCAL called for each set of pages.

Note: SCHLOCAL will default to the SCHEMA.CFG if the DRAWING.CFG is not found. To avoid problems either make sure that the hierarchical pages have a unique ".CFG" file with the same library setup as the top level or a SCHEMA.CFG file with the same library setup as the top level.

IX. HEXEDIT - Hexedecimal Editor

The HEXEDIT program will allow the user to review and edit any file, regardless of size in a word processing format. You may edit in string or hexidecimal mode. Block Move, Copy, Write, and Read are available. Place the cursor over a command you may have a question about and select Help. The online help will display a message about that command.

INVOKING HEXEDIT

HEXEDIT can be invoked from the Manager or DOS prompt. The format is:

HEXED FILENAME <Enter>

where FILENAME is the complete filename of the file to edit.

Note: The executable has a different name than the manager name due to the HEXED.HLP file is not an ASCII text file and can only be read by HEXED.EXE. The SCHEMA Manager Help Mode would not be able to read this file.

RUNNING HEXEDIT

When HEXEDIT loads, the screen is cleared and the HEXEDIT Main Menu, shown in figure 13, is displayed.

```
F1-Help  F2-BegBlk  F3-EndBlk  F4-CopyBlk  F5-WrtBlk  F6-ReadBlk  F7-SetBlk
File: \SCHEMA\SCHEMA.CFG                                        Byte: 0000000
┌──────────────────────────────────────────────────────────────────────────┐
 0000000  47 52 41 50  48 3D 50 32  2C 38 2C 32  36 2C 33 45   GRAPH=P2,8,26,3E
 0000010  2C 31 33 2C  34 2C 33 46  2C 33 46 2C  31 30 0D 0A   ,13,4,3F,3F,10▒
 0000020  50 52 49 4E  54 3D 5C 53  43 48 45 4D  41 5C 49 48   PRINT=\SCHEMA\IH
 0000030  50 4C 41 53  2E 44 52 56  0D 0A 50 4C  4F 54 54 45   PLAS.DRV▒▒PLOTTE
 0000040  52 3D 48 49  0D 0A 50 4C  4F 54 43 3D  31 0D 0A 50   R=HI▒PLOTC=1▒P
 0000050  58 4C 45 4E  3D 31 30 30  30 30 0D 0A  50 59 4C 45   XLEN=10000▒PYLE
 0000060  4E 3D 38 30  30 30 0D 0A  50 58 4F 46  46 3D 30 0D   N=8000▒PXOFF=0▒
 0000070  0A 50 59 4F  46 46 3D 30  0D 0A 50 45  4E 57 3D 31   ▒PYOFF=0▒PENW=1
 0000080  36 0D 0A 50  41 4E 3D 30  0D 0A 52 50  54 58 3D 30   6▒PAN=0▒RPTX=0
 0000090  0D 0A 52 50  54 59 3D 2D  31 36 0D 0A  42 41 43 48   ▒RPTY=-16▒BACH
 00000A0  55 50 3D 35  0D 0A 4D 4F  55 53 45 3D  31 2C 31 0D   UP=5▒MOUSE=1,1▒
 00000B0  0A 58 59 3D  4F 4E 0D 0A  43 47 52 49  44 3D 4F 4E   XY=ON▒CGRID=ON
 00000C0  0D 0A 56 47  52 49 44 3D  4F 4E 0D 0A  57 49 48 44   ▒VGRID=ON▒WIHD
 00000D0  52 3D 4F 4E  0D 0A 44 49  41 47 3D 4F  46 46 0D 0A   R=ON▒DIAG=OFF▒
 00000E0  56 4C 41 42  45 4C 3D 4F  4E 0D 0A 46  31 3D 53 43   VLABEL=ON▒F1=SC
 00000F0  48 45 4D 41  20 49 49 49  0D 0A 46 32  3D 31 30 30   HEMA IIII▒F2=100
 0000100  2D 30 30 30  31 2D 30 30  31 0D 0A 46  33 3D 31 41   -0001-001▒F3=1A
 0000110  0D 0A 46 34  3D 34 2F 31  30 2F 39 30  0D 0A 46 35   ▒F4=4/10/90▒F5
 0000120  3D 32 0D 0A  4C 49 42 52  31 3D 5C 53  43 48 45 4D   =2▒LIBR1=\SCHEM
└──────────────────────────────────────────────────────────────────────────┘
Alt: Save  Load  Quit    Mask  Create  Next  Prev
TAB-Hex  F8-Mode(HEX)   F9-Goto  F10-Search
```

Figure 13. HEXEDIT Main Menu

The command boxes list available HEXEDIT commands. To select a command, highlight the command using the mouse or the keyboard arrow keys and press **<Enter>** or click the mouse button.

Main Menu HEXEDIT Commands

F1 Help - Invokes a context sensitive help screen. If a command key has been entered before pressing F1, the help on the previous key entered is displayed in the edit window. Otherwise, the first page of the help file is displayed.

F2 Begin Block - Mark the cursor position from which the block to be marked begins. Dragging the mouse by pressing the left button or moving the cursor through arrow keys will highlight text being marked beginning from the cursor position where the block was marked. If cursor position is greater than start position, then the text is highlighted until the (cursor position - 1) location. If cursor position is less than start position, the text is highlighted from (cursor position + 1) location to start position.

F3 End Block - Mark the end of the block relative to cursor position. See also help on 'F2-BeginBlk' to determine the range of the block marked.

F4 Copy Block - Overwrites a block marked prior using F2 and F3, at cursor position.

F5 Write Block to Disk - Will write a block marked prior using F2 and F3 to a file. If the file being edited is "file.ext", then a filename "file.tmp" is created and the user is prompted to verify if the block can be written to the file "file.ext". The user is free to edit the file name and hit <Enter> or click the left mouse button to accept.

F6 Read Block from Disk - Will read the file specified by the user beginning at cursor position. If the file being edited is "file.ext", then a filename 'file.tmp', is created and the user is prompted to verify if the block can be written to the file "file.ext". The user is free to edit the filename and hit <Enter> or click the left mouse button to accept.

The block loaded overwrites the file being edited beginning with the current cursor location byte.

F7 **Set Block** - Set all bytes of a marked block to a specified value.

F8 **Mode(HEX/ASC/AUTO)** - Toggles the current mode. There are 3 modes for controlling the cursor.

(1) **AUTO:** In this mode the cursor can jump from HEX area to ASCII area when trying to cursor beyond a line using the left and right arrows. Trying to cursor beyond the right most ASCII column will re-position the cursor at the first column of the HEX area on the next line. Similarly, if a left arrow key is entered at first column in HEX area, the cursor is repositioned at the rightmost ASCII column on the line above.

(2) **HEX:** The cursor is permitted to move in the HEX area only.

(3) **ASCII:** The cursor is permitted to move in the ASCII area only.

If the current mode is Auto and cursor is in ASCII area, then pressing F8 will toggle the mode to HEX or Viceversa.

If the current mode is ASCII, then pressing F8 will change the mode to Auto and viceversa.

F9 **Goto** - This command will position the cursor at specified location. The input can be either a decimal number or a Hexadecimal number.

Decimal number format.... : NNNN
where N can take values 0-9

Hexadecimal number format : 0xNNNN or 0XNNNN , where N can take values 0-9, a-f or A-F

F10 Search - Search forwards from the current cursor location for a specified ASCII string or HEX sequence. If the search sequence is not found before the end of the file, a message is issued and the current cursor location is not changed. If the search sequence is found, the current cursor location is changed to that of the first byte of the search sequence.

Search backwards is invoked with ALT-F10 and looks for the search sequence between the current cursor location and the first byte of the file (address 0).

TAB TAB - Toggles cursor between ASCII area and HEX area maintaining the same file position.

ALT-S Save - Save changes to disk. A backup file is created with the name 'FILE.BAK' if the original file is "FILE.XXX" where ".XXX" is the original extension of the file being edited.

ALT-Q Quit - Discard changes and quit. No backup file is created.

ALT-L Load - Load a file specified.

ALT-M Mask or unMask - Mask or unMask an ASCII area.

ALT-C Create - Create a file specified.

ALT-N Next - Search forward (towards the end of file) for another occurence of the current search sequence.

ALT-P Prev - Search backwards (towards beginning of file) for a previous occurence of the current search sequence.

X. Command Macros

The SCHEMA Command Macros use the ANSI.SYS device driver that is supplied by DOS. A series of prompt statements define a key code that will invoke different commands within SCHDRAW.

INVOKING Command Macros

If you wish to use command macros, ANSI.SYS must be loaded into memory. Put the driver statement into the CONFIG.SYS file by adding the line:

DEVICE=C:\DOS\ANSI.SYS

The path to where ANSI.SYS is located may be different due to where it may be found on the hard disk. Reboot your system.

The Command Macros can now be invoked from the Manager or DOS prompt. The format is:

CMACROS <Enter>

RUNNING Command Macros

Enter SCHDRAW and now the Command Macros can be used. **<Alt>** key combination codes access commands in the Schematic Editor while **<Ctrl>** key combination codes access commands in the Object Editor.

<Alt> D	**DRAW-DOT**
<Alt> O	**DRAW-OBJECT**
<Alt> W	**DRAW-WIRE**
<Alt> C	**EDIT-COPY**
<Alt> E	**EDIT-ERASE**
<Alt> I	**EDIT-IMAGE**
<Alt> M	**EDIT-MOVE**
<Alt> R	**EDIT-REGION**
<Alt> L	**NOTES-LINE**
<Alt> S	**FILE-SAVE**
<F11>	**FILE-ED OBJ**
<Ctrl> B	**NAME-BODY**
<Ctrl> D	**NAME-DESC**
<Ctrl> I	**BODY-PIN**
<Ctrl> L	**BODY-LINE**
<Ctrl> P	**BODY-PARM n**
<Ctrl> T	**BODY-TEXT**
<Ctrl> S	**LIBR-SAVE**
<F12>	**LIBR-ED SCH**

Note: The **<Alt> C, E, M** commands work within the Object Editor.

These codes are defined in the CMACROS.BAT file. For more information about modification of these codes consult your MS-DOS manual about ANSI.SYS and the ASCII Key Codes.

XI. Printer Driver Source

The SCHEMA Printer Driver Source can be found on the distribution disk. The file is **DRIVESRC.ZIP**. The source is written in assembly and the driver were made using Microsoft Masm assembler.

Creating the Drivers using the MAKE File

Create a directory **\DRIVERS** off the root of your hard disk. Next create two subdirectories off the \DRIVERS root: **ORIG** and **MADE.**

Place the DRIVESRC.ZIP file in the \DRIVERS\ORIG directory. Unzip the file using PKUNZIP.EXE:

PKUNZIP \DRIVERS\ORIG\DRIVESRC

Next from the \DRIVERS\ORIG directory use the NMAKE program supplied with Microsoft C version 6. Type:

NMAKE <Enter>

All the drivers will be created in the **\DRIVERS\MADE** directory.

If you do not have the NMAKE program, then modify the MAKEFILE. file in the \DRIVERS\ORIG directory to suit the assembler of your choice.